Towards Automated 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 system for automating Semantic Role Labelling of Hindi-English code-mixed tweets. We explore the issues posed by noisy, user generated code-mixed social media data. We also compare the individual effect of various linguistic features used in our system. Our proposed model is a 2-step system for automated labelling which gives an overall accuracy of 84% for Argument Classification, marking a 10% increase over the existing rule-based baseline model. This is the first attempt at building a statistical Semantic Role Labeller for Hindi-English code-mixed data, to the best of our knowledge.

1 Introduction

Semantic Role Labelling (SRL) deals with identifying arguments of a given predicate or verb, in a sentence or utterance, and classifying them into various semantic roles. These labels give us information about the function played by the argument with respect to its predicate in the particular sentence.

With the growing popularity of social media, there is a lot of user generated data available online on forums such as Facebook, Twitter, Reddit, amongst many others. Subsequently, there is an increasing need to develop tools to process this text for its understanding. In multi-lingual communities, code-mixing is a largely observed phenomenon in colloquial usage as well as on social media. Code-mixing is described as *"the embedding of linguistic units such as phrases, words and morphemes of one language into an utterance of another language"* (Myers-Scotton, 1997). Social media data, Code-mixed text in particular, doesn't strictly adhere to the syntax, morphology or structure of any of the involved languages, which results in standard NLP tools not performing well with this data for a lot of tasks (Solorio and Liu, 2008; Çetinoğlu et al., 2016). T1 is an example from the corpus of Hindi-English code-mixed tweets (The Hindi words are denoted in italics).

T1 : "My life is revolving around '*bhook lagri hai*' and '*zyada kha liya*'"
Translation: My life is revolving around 'I am hungry' and 'I ate too much'

We present a 2-step system for automated Semantic Role Labelling of Hindi-English code-mixed tweets. The first step is to identify the arguments of the predicates in the sentence. The second step is to then classify these identified arguments into various semantic roles. We discuss the effect of 14 linguistic features on our system, of which 6 are derived from literature and rest are specific to Hindi or to the nature of code-mixed text. Semantic Role Labelling will aid in various NLP tasks such as building question-answering systems (Shen and Lapata, 2007), co-reference resolution (Ponzetto and Strube, 2006), document summarization (Khan et al., 2015), information retrieval (Moschitti et al., 2003; Osman et al., 2012) and so on.

The structure of this paper is as follows. We describe our data and the normalisation done for pre-processing of the text for our system in Section 2. The features used and compared are explained in detail in Section 3 along with the architecture of our system. We analyse the experiments and its results in Section 4. In Section 5, we conclude the paper.

2 Data and Pre-Processing

We used a dataset of 1460 Hindi-English code-mixed tweets comprising of 20,949 tokens labelled with their semantic roles (Pal and Sharma,

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 291–296

2019). This dataset is built on a dependency labelled corpus by Bhat et al. (2018). The tokens are parsed and labelled with Proposition Bank (PropBank) labels shown in table 1, depicting semantic roles of the arguments with respect to the predicates in the sentence (Palmer et al., 2005; Bhatt et al., 2009).

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: PropBank Tagset

Social media data doesn't conform to the rules of spelling, grammar or punctuation. These need to be taken into account to maintain uniformity for our system. We incorporated this in our preprocessing steps.

2.1 Misspelling

One of the most widely seen errors in social media data is 'typos', which are errors in spelling, usually slangs or typing errors. These errors can be broadly classified as follows:

- Misspelling leading to another word. For example, "thing"`[NN]`[1] misspelled as "think"`[VM]`.
- Omission of vowels - For example, the token "hr" is a commonly used abbreviation for the English word 'hour'. In our corpus, it referred to the Hindi word 'har' which is a quantifier and means 'every'.
- Elongation - tokens such as "Looooong", "Heyyyyy", *"pyaaaar"* and so on.
- Typing errors. For example, "saluet", which should have been 'salute'.
- Non-Uniformity in transliteration of Hindi tokens (usually written in Devanagari script) using the Roman alphabet. For example, the Hindi word for 'no' - *"nahi"* - had a lot of variation in its spelling in the corpus - 'nai', 'naee', 'nahi', 'nahee', 'nhi' etc.

We were able to detect some of the other errors through automated methods, such as elongation and some typing errors. Non-uniformity in transliteration was the most commonly found error in our corpus. These were all normalised and corrected manually to ensure a consistent spelling throughout the corpus.

2.2 Word Sense Disambiguation

A word can have different meanings according to the context in which it is used. T2 is an example from the corpus. The token "dikhny" refers to the Hindi verb *'xeKa'*[2] which means to look. This verb can have different senses according to its context as shown in table 2. From context we know the relevant roleset here would be `[xeKa.01]`. Available Frame files are used to identify rolesets for the verbs in the corpus (Vaidya et al., 2013; Bonial et al., 2014).

T2: "We are journilist and *hmy sechae dikhny se kiu rok ni skta*"
Translation: We are journalists and no one can stop us from seeing the truth.

Different senses for *xeKa*	
Roleset id	Meaning
`xeKa.01`	to see something
`xeKa.04`	to see (without volition)
`xeKa.06`	to show someone something
`xeKa.07`	used as a light verb

Table 2: Rolesets and meanings for the Hindi verb *xeKa*.

[1]Part of Speech (POS) tag

[2]WX notation

T3: "Shane on you *maine tuje pehle hi* Warne *kiya tha*"
Translation: Shane[NNP] on you, I had Warne[NNP] you before.
Implicit meaning: *Shame*[VM] *on you, I had warned*[VM] *you before.*

T3 is an interesting example from the corpus. The proper nouns 'Shane' and 'Warne' are used as the verbs 'shame' and 'warn' respectively in the sentence, due to their phonetic similarity. The speaker is possibly warning against the famous cricketer Shane Warne, and thus uses his name to convey the same. This sort of word play is not uncommon in social media data. These tokens are detected as proper nouns. We added them as predicates, according to their context, manually.

3 Semantic Role Labeller

Our Semantic Role Labeller has a 2-step architecture. The first step is a binary classification task wherein each token in the tweet is classified as 'Argument' or 'Not an Argument'. This step is called **Argument Identification**. In the second step, the identified arguments from the previous step are classified into the various semantic roles. This is called **Argument Classification**.

We used Support Vector Models (SVM) for binary classification. The identified arguments from this step are then classified into various semantic roles mentioned in Table 1. We used the Linear SVC class of SVM (Pedregosa et al., 2011) for one-vs-rest multi-class classification. The data was split in the ratio of 80:20 for training and testing respectively. All parameters of the LinearSVC were set to default for training.

3.1 Features used

Hindi and English have very different grammatical rules and vary greatly syntactically as well. We incorporated linguistic features in our system which may take into account these differences and help the labeller attain higher accuracy in identifying and classifying arguments.

3.1.1 Baseline Features

We used 6 baseline features which have been used extensively for the task of Semantic Role Labelling for English (Gildea and Jurafsky, 2002; Xue and Palmer, 2004). They are as follows:

- Predicate: Identified verb in the sentence
- Headword: Headword of the chunk
- HeadwordPOS: Part of Speech tag of the headword
- Phrasetype: Syntactic category of the phrase (NP, VP, CCP etc.)
- Predicate + Phrasetype
- Predicate + Headword

Semantic Arguments are identified at a phrase or chunk level. Hence we used features such as Headword of the chunk, phrasetype category, as baseline features. We also saw the impact of the part of speech (POS) tag of the Headword.

3.1.2 Features specific to Indian Languages

Previous work on Semantic Role Labelling have used the following features for Hindi specifically (Anwar and Sharma, 2016):

- Dependency(karaka relation): Paninian dependency label
- Named Entities
- HeadwordPOS + Phrasetype
- Headword + Phrasetype

We used the same features in our system. Named Entities have previously been seen to be a critical feature for Argument Identification task in English (Pradhan et al., 2004).

Vaidya et al. (2011) showed the strong corelation between Paninian dependency (karta) labels and Propbank labels for Hindi. This feature was also seen to give the best results for Hindi and Urdu monolingual corpus (Anwar and Sharma, 2016). Universal Dependencies (UD) have gained a lot of attention lately for cross-lingual parsing. Tandon et al. (2016) discussed and evaluated UD scheme for Hindi and also compared them to Paninian dependency labels. We evaluated UD part of speech(POS) tags and UD dependency labels as features in our system, as mentioned below.

- HeadwordPOS(UD) - UD part of speech tag of the headword
- UD dependency label

3.1.3 Features for code-mixed data

Since we are dealing with code-mixed text, we wanted to see the effect the identified language of a token may have. We thus used the following features:

- Predicate + language: Predicate and its identified language.
- Headword + language: The chunk headword and its identified language.

4 Results and Analysis

We do a thorough analysis of the individual features and their performance for the tasks of Argument identification and Argument Classification separately. Table 3 shows the precision, recall and F1 scores of the features for Argument Identification. Paninian Dependency labels give the highest F1-score of **78**.

Named Entities also give good results for Argument Identification. This is because Named Entities are usually arguments of a predicate. However, they by themselves don't capture much information about the role played by the argument in the sentence. Hence, the score for Argument Classification isn't that high, as can be seen in table 5.

Feature	Argument Identification		
	P	R	f-score
Predicate	33	50	40
Headword (HW)	52	47	49
HeadwordPOS	33	50	40
Phrasetype (PT)	41	34	37
Predicate-PT	42	65	51
Predicate-HW	55	49	51
Dependency	78	78	**78**
Named_Entity	57	50	65
HeadwordPOS-PT	41	34	37
Headword-PT	57	49	53
HeadwordPOS(UD)	32	50	39
UD_dependency	64	65	64
Predicate-language	43	65	52
Headword-language	55	47	51

Table 3: Individual feature performance for Argument Identification.

We also see a significant increase in accuracy when we use the combinational feature of predicate and its language, as compared to using only predicate as a feature (Table 3). T4 is an example from the corpus where the token "ban" is the Hindi verb `[bana]`, 'to become'. This can be confused with the English verb 'ban' (legal prohibition). In such cases, the language of the predicate token can play an important role.

T4: "Dear so called liberals, *kabhi* indian ***ban*** *ke dekho*"
Translation: Dear so called liberals, try being an Indian some time.

Feature	Argument Identification		
	P	R	f-score
Baseline	56	53	55
with predicate-lang	57	54	55
+dependency	81	76	78

Table 4: Accuracy scores for Argument Identification.

Table 4 gives the accuracy scores for the system using baseline features. Here, the score doesn't change much when we use 'predicate-language' as a part of our baseline. We are able to obtain the highest F1-score of 78 for this step by adding dependency label to our baseline features. The rule-based baseline model gives a much higher accuracy of 96.74% (Pal and Sharma, 2019). The baseline model uses the dependency tree structure of the sentence and identifies direct dependents of predicates as their arguments. Auxiliary verbs, post-positions, symbols, amongst others, are not considered as Arguments.

As the Classification step is based on the identified arguments from the first step, we chose to adopt a hybrid approach. We used the rule-based baseline system for Argument Identification, and used statistical approach with SVM for Argument Classification.

The precision, recall and F1 scores of the individual features for Argument Classification are given in Table 5. The best F1-score of 83 is again given by Paninian dependency labels. UD dependency gives a score of 80 which is slightly lower. Paninian dependency labels have performed better for both tasks as seen in Tables 3 and 5. There isn't much variation in performance between 'HeadwordPOS' and 'HeadwordPOS(UD)' for both steps.

The UD tagset is a coarser tagset. The UD POS tagset has only 17 tags, compared to the POS tagset developed for Indian languages which has

32 tags (Bharati et al., 2006). Similarly, in the Paninian dependency scheme, there are in total 82 relations, whereas UD has only 40. From the accuracy scores, we can infer that Paninian depepndency labels capture more semantic information than UD dependency labels.

Feature	Argument Classification		
	P	R	f-score
Predicate	06	09	06
Headword (HW)	18	10	13
HeadwordPOS	05	07	06
Phrasetype (PT)	08	10	08
Predicate-PT	05	08	06
Predicate-HW	05	06	06
Dependency	81	86	**83**
Named_Entity	20	14	16
HeadwordPOS-PT	07	09	08
Headword-PT	12	09	10
HeadwordPOS(UD)	08	11	09
UD_dependency	77	83	80
Predicate-language	06	10	07
Headword-language	18	11	14

Table 5: Individual feature performance for Argument Classification.

Feature	Argument Classification		
	P	R	f-score
Baseline	27	15	19
+*dependency*	84	84	**84**

Table 6: Accuracy scores for Argument Classification.

Table 6 gives the accuracy scores for Argument Classification while using baseline features, and after incorporating dependency labels. We obtained an F1 score of **84**. This is a significant improvement over the rule-based baseline model (Pal and Sharma, 2019) which gives an overall accuracy of 73.93% for Argument Classification.

5 Conclusion

In this work, we analyse the problems posed by code-mixed social media data. We present a system for automatic Semantic Role Labelling of Hindi-English code-mixed tweets. We used a hybrid approach of rule-based and statistical techniques for Argument Identification and Argument Classification respectively.

References

Maaz Anwar and Dipti Misra Sharma. 2016. Towards building semantic role labeler for indian languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4588–4595.

Akshar Bharati, Rajeev Sangal, Dipti Misra Sharma, and Lakshmi Bai. 2006. Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages. *LTRC-TR31*, pages 1–38.

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

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.

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

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

Atif Khan, Naomie Salim, and Yogan Jaya Kumar. 2015. A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing*, 30:737–747.

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. 1997. *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.

Ahmed Hamza Osman, Naomie Salim, Mohammed Salem Binwahlan, Rihab Alteeb, and Albaraa Abuobieda. 2012. An improved plagiarism detection scheme based on semantic role labeling. *Applied Soft Computing*, 12(5):1493–1502.

Riya Pal and Dipti Misra Sharma. 2019. A dataset for semantic role labelling of hindi-english code-mixed tweets. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 178–188.

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

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*.

Sameer S Pradhan, Wayne H Ward, Kadri Hacioglu, James H Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 233–240.

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)*, pages 12–21.

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.

Juhi Tandon, Himani Chaudhry, Riyaz Ahmad Bhat, and Dipti Sharma. 2016. Conversion from paninian karakas to universal dependencies for hindi dependency treebank. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 141–150.

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, 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.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 88–94.