A hybrid approach to Urdu verb phrase chunking

Wajid Ali
National University of Computer
and Emergence Sciences,
Lahore, Pakistan.
wajid.ali@msn.com

Sarmad Hussain
Center for Language Engineering,
Al-Khawarizmi Institute of Computer
Science, University of Engineering
and Technology, Lahore, Pakistan
sarmad.hussain@kics.edu.pk

Abstract

A variety of verb phrases exist in Urdu including simple verb phrases, conjunct verb phrases and compound verb phrases. This paper explains the structure of Urdu verb phrases, and details a series of experiment to automatically tag them. Initially, a rule based model is developed using 21 linguistic rules for automatic VP chunking. A 100,000 word Urdu corpus is manually tagged with VP chunk tags. The corpus is then used to develop a hybrid approach using HMM based statistical chunking and correction rules. The technique is enhanced by changing chunking direction and merging chunk and POS tags. The automatically chunked data is compared with manually tagged held-out data to identify and analyze the errors. Based on the analysis, correction rules are extracted to address the errors. By applying these rules after statistical tagging, further improvement is achieved in chunking accuracy. The results of all experiments are reported with maximum overall accuracy of 98.44% achieved using hybrid approach with extended tagset.

1 Introduction

Urdu is an Indo-Aryan language, spoken by more than 100 million speakers across the world. It is the national language of Pakistan and state language of India. Urdu has free phrase-order, i.e. the phrases within a sentence can arbitrarily change order[1], but the words within a phrase have a fixed order. As the order of the phrases is variable, the case markers (CM), which are explicitly written in Urdu as separate words, help determine the role of each phrase in a sentence. Verb Phrase (VP) is the head of a sentence and licenses the number as well as role of the other phrases in a sentence, e.g. subject, object, etc. The number of arguments licensed depends on the valency of the verb, also categorized as intransitive, transitive and di-transitive. This information is normally encoded in the subcategorization frame of a verb, which lists the number and type of arguments the verb licenses. Determining these phrases within a sentence is very useful for a variety of applications, and the process which directly labels these phrases is called chunking. Chunking helps to identify phrases in a sentence, which are further used for the development of natural language processing (NLP) applications like parsing, searching, machine translation, question-answering and information extraction. The current work focuses on chunking VP in Urdu.

Relevant Urdu VP analysis is summarized in Section 2. Section 3 presents some relevant chunking related work. Section 4 contains the detail of the tagged corpus developed for this task. Methodology is discussed in Section 5. The results and discussion are presented in Section 6. Section 7 concludes the work.

2 Verb phrases in Urdu

Minimally, an Urdu VP is represented by a single verb. However, a typical Urdu verb phrase contains a verb followed by one or more auxiliary verbs (AUX) and verb tense markers (VBT). Each is represented by a separate word. Some of the tense and aspect information is also encoded within the verb morphology (Hussain 2004). An Urdu verb phrase can be categorized into a simple verb phrase or complex verb phrase. In a complex verb phrase, the verb is formed by a combination of nominal + verb (called conjunct verb) or a verb + verb (called compound verb). These complex verbs are also referred as complex predicates.

[1] Though the meaning does not change, the phrase-order may change the emphasis of the phrase within a sentence.

Proceedings of the 8th Workshop on Asian Language Resources, pages 136–142,
Beijing, China, 21-22 August 2010. ©2010 Asian Federation for Natural Language Processing

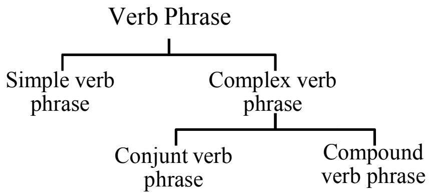

Figure 1: Categorization of Urdu verb phrase

The following subsections explain these types of verb phrases with examples.

2.1 Simple Verb Phrase

A simple verb phrase consists of a verb root followed by auxiliaries and tense verb, if any, as shown in (1).

(1)

علی سکول گیا[2]

Ali school *gaya*
Ali school went
"Ali went to school"

علی سکول جاتا ہے

Ali school *jata hai*
Ali school go *VBT-present*
"Ali goes to the school"

2.2 Complex Verb Phrase

A conjunct verb phrase consists of a nominal or an adjective and a verb, optionally followed by auxiliaries and tense verb, as shown in (2).

(2)

علی نے سبق یاد کیا

Ali ne sabaq *yaad kiya*
Ali *CM* lesson learn do-*past*
"Ali learnt the lesson"

علی نے کمرہ صاف کیا تھا

Ali ne kamraa *saaf kiya tha*
Ali *CM* room clean do-*past VBT-past*
"Ali cleaned the room"

A compound verb phrase consists of two verbs, optionally followed by auxiliaries and verb tense marker. The first verb is main verb and contributes the meaning of the sentence. The second verb adds additional information as shown in (3).

(3)

علی کام کر بیٹھا ہے

Ali kam *kar baeTha hai*
Ali work do sit VBT-*present*
"Ali has done the work"

Detailed analysis of Urdu VP is not in the scope of this paper. For further discussion, see e.g. Butt (1995) and Chakrabarti et al. (2008).

3 Earlier work on Chunking

The work on chunking based on machine learning was introduced by Church (1988) for English. Abney (1991) proposed the idea of parsing by chunks, defining the chunks in English by assuming that a chunk has syntactic structure. Chunking was used to convert sentences into non-overlapping phrases, like VP and Noun Phrase (NP), to parse the sentence. Chen (1993) proposed a probabilistic chunker based on Abney (1991). Ramshaw et al. (1995) used transformation based learning using a large annotated corpus for English. They proposed chunking as an IOB tagging task, where I marks the words which are *Inside* a chunk, O marks the words which are *Outside* the chunk and B marks the words which are at the *Beginning* of a chunk. Overall recall and precision achieved by this approach is about 88%.

Zhou et al. (2000) use standard HMM based tagging methods to model the chunking process, and achieved an accuracy of 91.99% precision and 92.25% recall using a contextual lexicon. Veenstra et al. (2000) use memory based phrase chunking with accuracy of 91.05% precision and 92.03% recall for English. Kudo et al. (2001) use support vector machines for chunking with 93.48% accuracy for English. Park et al. (2003) described a hybrid approach using rule based and memory based learning to chunk the phrases of Korean language. First, the rule based chunker is applied to chunk the phrases then memory based learning technique is used for the correction of errors which were not handled by rule based chunker. Grover et al. (2007) proposed rule based chunking using XML. They reported 90.18% precision and 92.49% recall for verb group chunker for English.

Singh et al. (2005) presented HMM based chunk tagger for Hindi. They divided chunk tagging into two main tasks: one was identification of chunk boundaries and the other was labeling of chunks. The Hindi annotated corpus of

[2] Urdu sentences are written from right to left.

200,000 words was used in their work. The data of 150,000 words used to train different HMM representations and 50,000 words data was kept aside as unseen data. The chunker was tested on 20,000 words and chunker achieved 92% precision with 100% recall for chunk boundaries by the HMM based chunker. Dalal et al. (2006) presented a maximum entropy based statistical approach to POS tagger and chunk tagger for Hindi. The model uses multiple features simultaneously to predict the tag for a word. The feature set is broadly classified as context-based features, word features, dictionary features and corpus-based features. The annotated corpus contained almost 35,000 words for training and testing. The reported accuracy was 87.4%. Agarwal et al. (2006) used Conditional Random Field for POS tagging and chunking Hindi text. Various experiments were carried out with various sets and combinations of features to mark a gradual increase in the performance of the system throughout the building process. A data of 21,000 words used for the training. The chunker gave 90.89% accuracy on the data for CONLL 2000.

4 Corpus and Tagset

For the current work, Part of speech (POS) tagged corpus containing 4,585 sentences and 101,414 words is used (from Muaz et al. 2009). Complex phrase is composed of a nominal, adjective or a verb combined with a *light verb*. In the POS tagged corpus used, light verbs are not tagged separately. However, tagging such verbs as light verbs helps determine whether the preceding word is part of verb phrase. So, we customized the tagset by introducing light verb tag (VBL) and infinitive light verb tag (VBLI) to better address the compound and conjunct verb phrases, following Sajjad (2007). The example demonstrates the light verb tag and chunk boundary of complex phrase.

علی <NNP> نے <PP> کمرہ <NN> صاف <JJ>
کیا <VBL> تھا <VBT>

علی<O><NNP> نے<O><PP> کمرہ<O><NN>
صاف <B><JJ> کیا <I><VBL> تھا <I><VBT>

Ali ne kamraa *saaf kiya tha*
Ali *CM* room clean do-*past VBT-past*
“Ali cleaned the room”

The IOB tagset is used to prepare chunk annotated data. The data of 3,650 sentences containing 81,430 words is for training, 530 sentences containing 9,985 words are used for analysis during the implementation of methodologies (as held-out data) and the remaining 405 sentences with 9,999 words are used for testing.

5 Methodology

A hybrid approach is used for VP chunking. First, a rule based chunker is developed for baseline. Then HMM based statistical approach is used. Finally, error correction rules are identified for further correction. The methodology is described below.

5.1 Rule Based Chunking

Initially, a set of 21 hand crafted rules are derived, based on experience through manual tagging, for VP chunking. These rules are incrementally built and applied using the training corpus.

5.2 Statistical Chunking

A statistical model for automatic tagging is also developed. Given a sequence of *n* words, there are corresponding t_1 to t_n POS tags and c_1 to c_n chunk tags. The aim is to find the most probable chunk sequence for given the POS tags.

$$\hat{C} = arg \max P(t_1^n/c_1^n) \,.\, P(c_1^n)$$

We assume that the probability of a POS tag depends on its own chunk tag and the probability of a chunk tag is dependent only on the previous two chunk tags. Using chain rule, problem is reduced to the following equation.

$$\hat{C} = arg \max \prod_{i=1}^{n} P(t_i/c_i).P(c_i/c_{i-1}, c_{i-2})$$

TnT tagger (Brants 2000) is used for training and testing, which is based on this model. All the experiments are executed using its default option of second order HMM (trigram model, as presented).

5.3 Error Correction Rules

The statistical tagger is run on the held out data and errors are analyzed to derive rules to fix them as part of the post-processing module. Based on error analysis, twelve rules are identified. Here we discuss some errors and corresponding rules for correction. The complete list of error correction rules is included in the appendix. The most frequent error was assigning I tag to VBT, when it was not preceded by a verb, as it is itself the verb in this case. In this case, it should have been assigned B tag, as it is the beginning of the verb phrase. For example, see the tags underlined below.

<O> <PP> کی <O> <NN>بچوں <O> <JJ>زیرتعلیم
<I> <VBT> ہے <O> <CD>۹۱۹<O> <CD> تعداد

Zair-e-taleem bachoon ki tadaad 919 hai
Under-education children’s number 919 is
“Number of children under education is 919”

The following simple rule makes the needed correction.

- If $POS(w_i) = VBT$ and $POS(w_{i-1})\,! = \{VB, VBI, VBL, VBLI, AUX\}$, then chunk tag for w_i is B.

Another error is to assign O tag to JJ while it precedes the light verb and follows NN. Here it should be the part of the verb phrase.

<O> <NN> جیلیں <O> <PP> کی <O> <NN> بچوں
<I> <VBT> گی <B> <VBL> ہوں <O> <JJ> علیحدہ

Bachon ki jailain alaidha hon gi
Children’s jails separate be *VBT-future*
“Children’s jails will be separate”

The following rule makes the correction.

- If $POS(w_i) = VBL$, $POS(w_{i-1}) = JJ$ and $POS(w_{i-2}) = NN$, Then chunk tag for w_{i-1} is B and w_i is I.

B tag is assigned to VBL while it follows the CVRP and CVRP follows VB. Here it should be the part of same verb phrase starting from VB.

کر <B> <VB> لکھ <O> <NN> خط <O> <PRP> وہ
<I> <VBT> گیا <B> <VBL> آ <I> <CVRP>

Wo khaat likh ker aa gya
He letter write do come *VBT-past*
“He came after a writing letter”

The following rule makes the correction.

- If $POS(w_i) = VB$, $POS(w_{i-1}) = CVRP$ and $POS(w_{i-2}) = VBL$, Then chunk tag for w_i is I.

One more error is to assign O tag to WALA while it follows VBLI. Here it should be the part of the verb phrase.

والی <I> <VBLI> کرنے <B> <NN> کام
<O><PP> نے <O><NN> خاتون <O><WALA>
<B><VB> بتایا

kaam karnay wali khaton ne batayaa
Work doing WALA[3] women CM told
“Working woman told”

The following rule makes the correction.

- If $POS(w_i) = WALA$, $POS(w_{i-1}) = VBLI$, Then chunk tag for tag for w_i is also I.

5.4 Architecture of VP Chunker

A POS tagged sentence is the input of the VP chunker. The input data is prepared in a specific format and each line contains only a POS tag corresponding to the word in the sentence. TnT Tagger outputs appropriate chunk tag against each POS tag using HMM model. Then post processing is performed on the output of the statistical chunker to enhance the accuracy by applying the error correction rules. Figure 2 shows the architecture of this VP chunker.

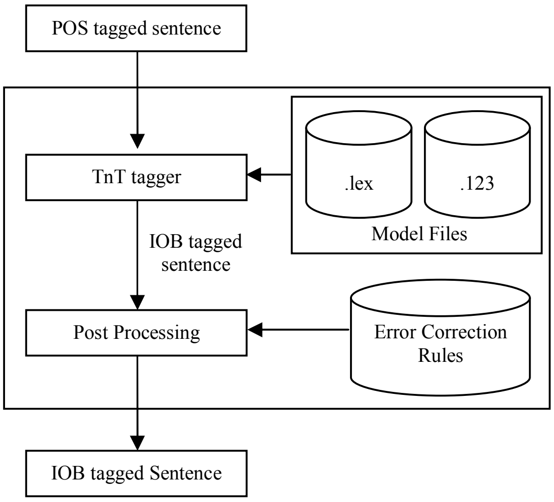

Figure 2: Architecture of VP Chunker

[3] There is no easy translation for WALA. See Muaz and Khan (2009).

5.5 Experiments

VP chunking system is evaluated by conducting a series of experiments. The data is trained and tested using TnT tagger. Three additional factors are used. First, if one scans the sentence in reverse order, one may be able to better predict phrase boundary, as CM comes at the end of a NP. Thus, both Right to Left (default for Urdu) and Left to Right (reverse) directions are explored for scanning and tagging. Second, IOB tagging scheme is further fine-grained by merging it with POS tagset, as an alternate system. Thus B-NN and B-VB are used as different tags, instead of just using B. Third, only statistical vs. hybrid methodologies are used. So, a total nine experiments including rule based model (as baseline) are performed which are listed in Table 1.

Table 1: Scheme for VP chunking experiments

No.	Tagset	Model	Scanning
1.	IOB	Rule Base	Right to left
2.	IOB	Statistical	Right to left
3.	IOB	Hybrid	Right to left
4.	IOB Extended	Statistical	Right to left
5.	IOB Extended	Hybrid	Right to left
6.	IOB	Statistical	Left to right
7.	IOB	Hybrid	Left to right
8.	IOB Extended	Statistical	Left to right
9.	IOB Extended	Hybrid	Left to right

6 Results and Discussion

6.1 Results

There are a total of nine experiments which are performed. First Rule based method is executed on testing data using 21 handcrafted linguistic rules for automatic VP chunking and 93.23% accuracy is achieved. Then statistical experiment is executed on same testing data with simple IOB tagset scheme, and right to left scanning direction. The precision and recall for I, O and B tags are calculated separately. The overall accuracy is 95.14%. By applying error correction rules on this output of statistical chunker, we obtain an overall accuracy of 98.14%. This is given in Table 2 below.

Experiments are also performed using extended tagset by merging IOB tag with POS tag. The overall accuracy of experiment is improved to 95.95%. Error correction rules are applied on output of the statistical chunker, and accuracy is improved to 98.44%.

When the scanning direction is changed to Left to Right, the overall accuracy of statistical approach with simple tagset is 95.06% and 98.02% overall accuracy is obtained using hybrid approach. When extended tagset is used with statistical and hybrid approaches in this scanning direction, the overall accuracy of 95.86% and 98.29% is achieved respectively.

Table 2: Results of VP chunking Experiments

No.	Methodology	Over all result (%)	B-tag		I-tag		O-tag	
			Precision	Recall	Precision	Recall	Precision	Recall
1.	Rule Base (all rules) and RTL scanning	93.23	94.93	56.78	92.64	82.95	99.94	92.99
2.	Statistical using IOB tagset and RTL scanning	95.14	82.52	75.08	88.45	85.17	97.54	99.22
3.	Hybrid using IOB tagset and RTL scanning	98.14	96.21	87.54	92.00	96.62	99.42	99.72
4.	Statistical using Extended tagset and RTL scanning	95.95	83.98	79.13	88.88	86.41	98.27	99.51
5.	Hybrid using Extended tagset and RTL scanning	98.44	97.16	90.07	93.10	96.62	99.45	99.77
6.	Statistical using IOB tagset and LTR scanning	95.06	81.64	74.77	88.20	84.93	97.62	99.22
7.	Hybrid using IOB tagset and LTR scanning	98.02	95.95	86.32	91.57	96.62	99.28	99.69
8.	Statistical using Extended tagset and LTR scanning	95.86	83.24	78.01	88.45	85.67	98.78	99.51
9.	Hybrid using Extended tagset and LTR scanning	98.29	96.80	88.75	91.78	96.62	99.42	99.74

6.2 Discussion

The aim of this research has been to develop an automatic verb phrase chunker for Urdu. To get maximum accuracy different experiments have been conducted using rule base, statistical and hybrid approaches. The intention has been to identify the factors which are important for high accuracy. The experiments show that statistical technique performs better than the rule based system, though the accuracy of the rule based system may be increased further by adding more rules to the repository, which is a tedious process. It is also observed that a few simple error correction rules give a significant 3% improvement in accuracy. Moreover, merging POS tag with IOB tag gives minor improvement in accuracy but reversing scanning direction decreases accuracy.

These results are comparable, even a bit better than the work reported for English. The results are also comparable, perhaps a little better, than Hindi, as reported in the literature, even though Hindi is same as Urdu as spoken. Though the difference in results from English can be attributed to the grammatical differences, it is interesting to note the differences with Hindi. Future work should explore how much of the difference can be attributed to the difference in data used for training, and how much of this difference is caused due to a slight morpho-syntactic difference between the two languages, where in Hindi the case markers are written with the noun as single word in Devanagari script, but are written as separate words from nouns in Urdu using Arabic script.

7 Conclusion

In this paper, we have proposed a hybrid approach to learn verb phrase chunking for Urdu using HMM based statistical chunking and rule based correction afterwards. We have performed different experiments to get maximum accuracy and found the scheme based on hybrid approach with extended tagset and right to left scanning gives the best accuracy of 98.44%.

Appendix

The rules for verb phrase chunking are as following:

1. If $POS(w_i) = VBT$ and $POS(w_{i-1})\ ! = \{VB, VBI, VBL, VBLI, AUX\}$, Then chunk tag for w_i is B.
2. If $POS(w_i) = VB$ and $POS(w_{i-1}) = \{VB, VBI, VBL\}$, Then chunk tag for w_i is I.
3. If$POS(w_i) = VB$, $POS(w_{i-1}) = CVRP$ and $POS(w_{i-2}) = VBL$, Then chunk tag for w_i is I.
4. If $POS(w_i) = VBL$, $POS(w_{i-1}) = JJ$ and $POS(w_{i-2}) = NN$, Then chunk tag for w_{i-1} is B and w_i is I.
5. If $POS(w_i) = VBL$, $POS(w_{i-1}) = NN$ and $POS(w_{i-2}) = JJ$, Then chunk tag for w_{i-1} is B and w_i is I.
6. If $POS(w_i) = VBLI$, $POS(w_{i-1}) = NNP$ and $POS(w_{i-2}) = NN$, Then chunk tag for w_{i-1} is B and w_i is I.
7. If$POS(w_i) = VBLI$, $POS(w_{i-1}) = NN$ and $POS(w_{i-2}) = NNP$, Then chunk tag for w_{i-1} is B and w_i is I.
8. If $POS(w_i) = WALA$, $POS(w_{i+1}) = NN$ and $POS(w_{i+2}) = NN$, Then chunk tag for w_{i+1} is I.
9. If$POS(w_i) = WALA$, $POS(w_{i+1}) = JJ$ and $POS(w_{i+2}) = NN$, Then chunk tag for w_{i+1} is I.
10. If $POS(w_i) = WALA$, $POS(w_{i-1}) = VBI$, Then chunk tag for w_i is I.
11. If $POS(w_i) = WALA$, $POS(w_{i-1}) = VBLI$, Then chunk tag for w_i is I.
12. If $POS(w_i) = WALA$, $POS(w_{i-1}) = VBLI$, Then chunk tag for tag for w_i is also I.

References

Abney S. 1991. *Parsing by Chunks: Principle based parsing*. Kluwer Academic Publishers, Dordrecht.

Agarwal H. and Mani A. 2006. *Part of Speech Tagging and Chunking with Conditional Random Fields*. In proceedings of NLPAI Machine Learning Context, Mumbai, India.

Butt M. 1995. *The Structure of Complex Predicates in Urdu*. Stanford, CA: CSLI Publications.

Brant T. 2000. *TnT: a statistical part of speech tager*. In proceeding of the sixth conference on applied natural language processing, Seattle, Washington: 224–231.

Chakrabarti D., Mandalia H., Priya R., Sarma V., and Bhattacharyya P. 2008. *Hindi Compound Verbs and their Automatic Extraction*. Computational Linguistics (COLING08), Manchester, UK.

Chen Kuang-Hua and Chen Hsin-His. 1993. *A Probablistic Chunker*. In proceedings of ROCLING VI.

Church K. 1988. *A stochastic parts program and noun phrase parser for unrestricted text*. In proceedings of Second Conference on Applied Natural Language Processing: 136–143.

Dalal A., Nagaraj K., Sawant U. and Shelke S. 2006. *Hindi Part-of-speech tagging and chunking: A Maximum Entropy Approach*. In proceedings of NLPAI Machine Learning Context, Mumbai, India.

Grover C. and Tobin R. 2007. *Rule Based Chunking and Reusability*. In proceedings of the Fifth international conference on Language Resources.

Hussain, S. 2004. *Finite-State Morphological Analyzer for Urdu.* Unpublished MS Thesis, National University of Computer and Emerging Sciences, Lahore, Pakistan. Available at www.crulp.org.

Kudo T. and Matsumoto, Y. 2001. Chunking with Support Vector Machines. Proceedings of NAACL 2001: 1013–1015.

Muaz A., Ali A. and Hussain S. 2009. *Analysis and Development of Urdu POS Tagged Corpora*. In proceedings of the 7th Workshop on Asian Language Resources, IJCNLP'09, Suntec City, Singapore.

Muaz, A., Khan, A. 2009. "The Morphosyntactic Behavior of 'Wala' in Urdu Language", In the *Proceedings of 28th Annual Meeting of the South Asian Language Analysis Roundtable, SALA'09*, University of North Texas, USA. Available at http://www.crulp.org.

Park S. and Zhang B. 2003. *Text Chunking by Combining Hand-Crafted Rules and Memory-Based Learning*. In proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1: 497–504.

Ramshaw L. A. and Marcus M. P. 1995. *Text chunking using transformation based learning*. In proceedings of the third ACL workshop on Very Large Corpora, Somerset, NJ: 82–94.

Sajjad, H. 2007. *Statistical Part of Speech Tagger for Urdu.* Unpublished MS Thesis, National University of Computer and Emerging Sciences, Lahore, Pakistan. Available at www.crulp.org.

Singh A., Bendre S. and Sangal R. 2005. *HMM Based Chunker for Hindi*. In the Proceedings of IJCNLP-05: The Second International Joint Conference on Natural Language Processing.

Veenstra, J. and van den Bosch, A. 2000. *Single-Classifier Memory-Based Phrase Chunking*. In proceedings of CoNLL-2000 and LLL-2000: 157–159.

Zhou, G., Su, J. and Tey, T. 2000. *Hybrid Text Chunking*. In proceedings of CoNLL- 2000 and LLL-2000: 163–165.