A Tagged Corpus and a Tagger for Urdu

Bushra Jawaid, Amir Kamran, Ondřej Bojar

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské nám. 25, Praha 1, CZ-118 00, Czech Republic
{jawaid,kamran,bojar}@ufal.mff.cuni.cz

Abstract

In this paper, we describe a release of a sizeable monolingual Urdu corpus automatically tagged with part-of-speech tags. We extend the work of Jawaid and Bojar (2012) who use three different taggers and then apply a voting scheme to disambiguate among the different choices suggested by each tagger. We run this complex ensemble on a large monolingual corpus and release the tagged corpus. Additionally, we use this data to train a single standalone tagger which will hopefully significantly simplify Urdu processing. The standalone tagger obtains the accuracy of 88.74% on test data.

Keywords: large monolingual data, part-of-speech tagger, Urdu annotated data

1. Introduction

Despite the large number of speakers, Urdu is a resource-poor language when it comes to finding core language processing modules such as morphological analysers, taggers, parsers and so on. We focus on the very first step, obtaining a reasonably accurate but primarily reliable and easy-to-deploy Urdu tagger.

Jawaid and Bojar (2012) made an attempt to use existing taggers of Urdu for annotating a test corpus and observed that the best accuracy was obtained using a rather peculiar combination of existing resources: a manually tagged corpus by CRULP (Center for Research in Urdu Language Processing)[1], and two taggers: (1) morphological analyser of Humayoun (2006), called HUM analyzer in the following, and (2) tagging module of Urdu shallow parser developed by Language Technologies Research Center of IIIT Hyderabad[2], called SH parser in the following. Jawaid and Bojar (2012) use the corpus by CRULP to train a SVM tagger by Giménez and Márquez (2004), obtaining a third tagger.

To annotate the final test corpus, Jawaid and Bojar (2012) run all the three taggers, convert their output into common representation, unify their different tagsets[3] and apply a voting scheme on the combined output of all three taggers to pick the tag with the maximum votes. The maximum accuracy they obtained on the voted data is 87.98%.

In this paper, we try to overcome the practical issues of the setup by Jawaid and Bojar (2012): the need to install and operate three different taggers, output converters and a final voting script. We do so by applying this complex pipeline to a relatively large corpus and then training a "one-shot" tagger from this corpus.

Once all final checks are done, we release both the large automatically annotated corpus as well as the trained one-shot tagger.

[1] http://www.crulp.org/software/ling_resources/UrduNepaliEnglishParallelCorpus.htm

[2] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

[3] For the SVM tagger, they actually map the CRULP training data to the unified tagset before training the SVM tagger, so the SVM tagger already operates in the desired tagset instead of cumulating errors of tagging with errors of a subsequent mapping.

2. Urdu Monolingual Data Resources

Emille Project (Enabling Minority Language Engineering) is the first one to take the initiative for collecting South Asian language resources (Baker et al., 2002). Besides publishing parallel corpus that contains translations of English into several Indic languages such as Hindi, Urdu, Bengali, Gujarati, and Punjabi, Emille has also published two different monolingual corpora for all covered languages: a spoken corpus and a written corpus.

The Urdu spoken corpus consists of 512,000 (approximately 0.5M) words of Spoken Urdu, transcribed from broadcasts of BBC Asian Network and BBC Radio. The Urdu written corpus (incorporated from the CIIL Corpus) contains 1,640,000 (approximately 1.6M) words. These Urdu resources are also manually annotated with a morpho-syntactic tagset Hardie (2003).

3. Data: Our Urdu Corpus

In this section we give an overview of monolingual data we use in this work.

In 2014, we crawled several websites and gathered a large corpus of Urdu text. The collected corpus is a (unlabeled) mix of the following major domains: News, Religion, Blogs, Literature, Science, Education and numerous others. The data is crawled from following web sources: BBC Urdu[4], Digital Urdu Library[5], Minhaj

[4] http://www.bbc.co.uk/urdu/

[5] http://www.urdulibrary.org/

Books[6], Faisaliat[7], Awaz-e-Dost[8] and Urdu Planet[9].

The remaining two data sources, Noman's Diary[10] and iFastnet[11] were collected in 2010. Table 1 shows the detailed statistics of the collected corpora. The resulting monolingual corpus has around 95.4 million tokens distributed in around 5.4 millions sentences.

These figures present the statistics of all the domains whose data we annotate using our joint tagger and release afterwards.

Corpus	Sentences	Tokens	Vocabulary
Urdu Planet	4,793,736	78,045,722	536,789
BBC Urdu	423,828	11,974,394	96,008
Urdu Library	96,240	1,692,948	44,812
Minhaj Books	83,282	2,458,402	39,955
iFastnet	24,639	427,324	28,103
Awaz	22,031	388,498	20,591
Noman's Diary	18,664	375,531	19,770
Faisaliat	2,155	49,008	5,542
Total	5,464,575	95,411,827	582,795

Table 1: Statistics of our Urdu data after deduplication.

This corpus represents the first main result of this work and it is publicly available online[12]. The corpus can be used for academic research purposes only.

3.1. Data Crawling

Our web crawler is based on Selenium[13] that provides advanced features to interact with web pages and provides the functionality to extract text from specific sections of a web page. This feature helped us to extract pure Urdu contents from web pages, stripping HTML tags and also removing other unrelated text.

For all the web sources except Urdu Planet, we provided the crawler with specific HTML and CSS selection criteria used to mark sections of the pages that contain the relevant Urdu contents. The crawler then extract the data only from the selected HTML sections and discards the rest.

Urdu Planet is our largest extracted corpus. Compared to other data sources, it suffers from the highest level of noise because it is a repository of large number of blogs each with different HTML structure and we were unable to come up with a generic HTML pattern to extract filtered data from this source.

[6] http://www.minhajbooks.com/urdu/control/Txtformat/

[7] http://shahfaisal.wordpress.com/

[8] http://awaz-e-dost.blogspot.cz/

[9] http://www.urduweb.org/planet/

[10] http://noumaan.sabza.org/

[11] http://kitaben.ifastnet.com/

[12] http://hdl.handle.net/11858/00-097C-0000-0023-65A9-5

[13] Selenium is a browser automation toolkit that is used primarily for automated web application testing.

3.2. Data Cleaning

The plain text obtained from our crawler is further cleaned in the following steps:

- All duplicated paragraphs are discarded and only a unique occurrence of each of them is kept.
- A language detection tool[14] is used to discard sentences in a foreign language (data sources of religious domain such as Minhaj Book contains several sentences in Arabic and their English translation whereas rest of the sources contain mix of foreign sentences in English, Arabic and some regional Pakistani languages).
- Data is tokenized using a simple tokenization script by Dan Zeman. The script replaces any control and space characters (including TAB and non-breaking space) by spaces and surrounds punctuation with spaces.
- After tokenizing the data, the paragraphs are split into sentences using Urdu sentence end markers such as full stop and question mark.
- Only sentences with two or more tokens are kept.
- Because each corpus contains numerals written in various styles (mainly Western Arabic as in English but also Eastern Arabic and its variants), we normalized all data sources by converting all numerals to the Western Arabic variant.
- Noise from the data is removed by discarding the tokens such as white stars, left or right arrows, smiley icons, bullets etc. Invalid UTF-8 characters are also removed.
- Foreign sentences (including those not recognized by our language detection tool) such as Sindhi, one of the regional language of Pakistan, are deleted. Up to 50% of the tokens in the sentence are allowed to come from English. Sentences with more foreign words are removed.
- Some checks focus on white space: sentences with no spaces between tokens are deleted. On the other hand, spaces are inserted (if missing) between English and Urdu words.
- Bracketed English phrases in the middle of sentence are also removed because they mostly represent some translation or explanation that should not be part of a natural Urdu sentence.

All the scripts and tools that are used for data cleaning are also released with the data.

[14] The language detection tool was developed by the main author during her Master's thesis.

HUM Analyser	UNK|؟|؟ ...|N+NF+Pl+Voc+Masc|N+NF+Sg+Nom+Masc|InterPron3+IntPF3+Pl+Masc|کیسے|کیسا
SH Parser	SYM+punc++++++poslcat=NM|?|? WQ+adv++++++poslcat=NM|کیسے|کیسے
SVM Tagger	SM_1|؟ ADV_0.713392372067598-AKP_0.163102676350609-KP_0.123504951581793|کیسے

Table 2: The output of HUM Analyzer, SH Parser and SVM Tagger converted to the common format.

4. Processing: Three POS Taggers Consolidated

To automatically provide part-of-speech tagging for our corpus, we use the following three taggers: HUM analyser, SH parser and the SVM tagger trained by Jawaid and Bojar (2012).

We first annotate our monolingual corpus described in Section 3. using all the three taggers. Table 2 shows the output of all three taggers.

We convert the output of HUM analyser and SH parser to a "|" delimited format, discarding lemma and stripping all the morphological analysis from the annotations. We keep only the word form and POS tag. The output of SVM tagger is also constructed in the required format.

Each of the taggers uses its own tagset, and we follow Jawaid and Bojar (2012) who unify these tagsets to a single tagset[15] proposed by Sajjad and Schmid (2009). The only exception is the output of SVM tagger that does not need to be unified again because the tagger is trained on unified CRULP's manually annotated data.

HUM Analyser	SM|؟ AKP|کیسے
SH Parser	SM-PM-EXP|? KD-KP-QW|کیسے
Concatenated	ADV_0.713392372067598-|کیسے AKP_0.163102676350609- KP_0.123504951581793+KD- SM-PM-|؟ KP-QW+AKP EXP+SM+SM_1

Table 3: Unified output of HUM Analyzer and SH Parser on Sajjad's tagset and the Concatenated output of all three taggers.

Table 5 shows the mapping from the individual tagsets to the unified tagset (called Sajjad's tagset in following).

We unify the output of HUM analyser and SH parser on Sajjad's tagset and concatenate the unified output of both taggers with the SVM tagger output, as shown in Table 3. SH parser faces a problem of converting Urdu punctuations into their English variants, such as replacing end of sentence marker with full stop. That is why, in Table 3 we see Western style question marker (?) instead of its Arabic variant (؟). On concatenated output, we apply the voting scheme and afterwards fallback options, the setup known to give the best accuracy in Jawaid and Bojar (2012). Table 4 shows the concatenated output of all taggers after applying voting and static fall back options.

The final corpus after applying voting and fallback strategy is used for training the standalone SVM tagger.

After Voting	VB|چنگھاڑتی CC|اور MUL+VB+NN|چیختی
After Fallback	VB|چنگھاڑتی CC|اور VB|چیختی

Table 4: Concatenated output of all three taggers after applying voting and fallback options.

5. Our Standalone Tagger

The second main result that is accompanying our submission is a standalone tagger trained on the automatically tagged corpus.

We use SVM Tool (Giménez and Márquez, 2004) for the training of the standalone tagger. Sajjad and Schmid (2009) show a comparison of four state-of-the-art probabilistic taggers (TnT tagger, TreeTagger, RF tagger and SVM tool) for Urdu and report that on a training corpus of 100,000 tokens, SVM tool outperformed the other taggers.

SVM tool comes with the implementation of five different kinds of models for training. We use 'model 4' with tagging direction from right-to-left. Compared to other models, the model 4 helps in learning more realistic and refined model by artificially marking some of the words as unknown at training time. Jawaid and Bojar (2012) and Sajjad and Schmid (2009) used the same model for training their SVM taggers.

The test corpus for tagger evaluation consists of tagged 8K tokens[16] from BBC News[17]. The same test set is used for evaluation by Jawaid and Bojar (2012) and Sajjad and Schmid (2009) in their work. We call it Sajjad's test data in the following.

For training, the monolingual data is combined into two different settings: with and without BBC corpus. The reason of training the tagger on data excluding BBC corpus is to do the fair evaluation of tagger because Sajjad's test data also comes from BBC Urdu. However, due to time constraint we only train the tagger on data without BBC corpus for now but, we plan to release the final tagger trained on entire monolingual data (except Urdu Planet). Table 6 shows the

[15] http://www.cle.org.pk/Downloads/langproc/UrduPOStagger/UrduPOStagset.pdf

[16] The test set is tagged using Sajjad's tagset and it is freely available online: http://www.ims.uni-stuttgart.de/~sajjad/resources.html

[17] http://www.bbc.co.uk/urdu/

Sajjad's Tagset	CRULP	HUM Analyzer	SH Parser
A	JJRP	PostP, Part	PSP
AA	AUXA	—	VAUX
AD	—	RelPron2	DEM
ADJ	JJ	Adj, Adj1, Adj2, Adj3, AdjD	JJ, XC
ADV	RB, I	Adv	RB, INTF, NST
AKP	—	InterPron1, InterPron2, InterPron3	—
AP	—	RelPron2	NST
CA	CD	Num	QC, ECH
CC	CC	Conj	CC
DATE	DATE	—	—
EXP	SYM	—	SYM
FR	FR	—	—
G	PRP$	PossPron	PRP
GR	PRRFP$	—	PRP
I	ITRP	Part	RP
INT	INJ	Intjunc	JJ
KD	—	InterPron	WQ
KER	KER	PossPostPos	PSP
KP	—	InterPron	WQ
MUL	MUL	Verb, Verb1	VM
NEG	NEG	Neg	NEG
NN	NN, NNCM, NNC, NNCR, MOPE, MOPO, NNL	N	NN, XC
OR	OD	RelPron2, N	QO
P	CM	PossPostPos	—
PD	DM	DemPron	PRP
PM	PM	—	SYM
PN	NNP, NNPC	PN	NNP, XC
PP	PR	PersPron, RelPron1	DEM
Q	Q	IndefPron1, IndefPron2, RelPron2, IndefPron, RelPron3	QF
QW	QW	Quest	WQ
RD	DMRL	RelPron	—
REP	PRRL	RelPron	PRP
RP	PRRF	RefPron	PRP
SC	SC	Conj	CC
SE	SE, RBRP	PostP	PSP
SM	SM	—	SYM
TA	AUXT	—	VAUX
U	U	—	—
UNK	UNK	UNK, Verb3, Verb_Aux	UNK
VB	VB, VBL, VBI, VBLI, VBT	Verb, Verb1, Verb2	VM
WALA	WALA	—	—

Table 5: Tagset mapping of Humayoun Morphological Analyzer, Urdu Shallow Parser and CRULP tagset to a common Sajjad's tagset.

statistics of the monolingual data used for the training of the tagger. The row "Unknown tokens" in Table 6 shows that the 105 test tokens were never seen in the training data with BBC and 165 tokens were never seen in the training data without BBC. Similarly, "Unknown Types" shows the unique word count of missing test tokens in the training data.

We train SVM tool on the voted data. The voted data is created using the voting setup that reaches the highest accuracy, i.e. 87.73% correctly tagged tokens of the test corpus. The implementation of the voting strategy in Jawaid and Bojar (2012) that produces the best accuracy setup is as follows: each tagger has the power of 1 vote. If the tagger emits more than one tags for a token, this one vote is split uniformly among all the suggested tags. We take the top 3 options from SVM and normalize their probabilities to sum to the one vote of SVM. In those cases where SVM predicts only 1 tag to express its certainty, we give it a preference over other taggers, i.e. no voting in this case. Votes for the unknown tag (UNK) are discarded.

The tag that receives the highest sum of votes is selected. In case of a voting conflict, i.e. two or more tags receive the same number of votes, we resolve the ambiguity using a static preference list.

6. Evaluation

Jawaid and Bojar (2012) give the comparison of accuracies of individual taggers and the different voting setups. Table 7 shows the overall accuracy of tagger when trained on data without BBC corpus. We also show the tagger accuracy on known and unknown words. We see that the standalone tagger does not decrease the performance at all. In fact, it helps a tiny little bit. Only 45% of the unknown tokens are correctly tagged by our tagger and that is rather lower than expected. The accuracy on test data without using the stand-alone tagger is also presented in Table 7. Table 8 shows the confusion matrix for the most confused tag pairs.

	Training Data		Test Data
	With-BBC	Without-BBC	
Sentences	670,847	247,019	404
Tokens	17,312,155	5,383,519	8,670
Types	141,608	81,285	1,917
Unknown Tokens	105	168	-
Unknown Types	60	92	-

Table 6: Statistics of test data and training data with and without BBC corpus.

	Accuracy		
	Total	Known	Unknown
Complex Ensemble	87.73%	-	-
Standalone Tagger	87.80%	88.46%	54.76%

Table 7: Overall accuracy of tagger on test data and also individual accuracy of Known and Unknown words. Accuracy without tagger is obtained after running complex ensemble of taggers on test data and applying voting scheme afterwards.

Proper nouns (PN) in Urdu do not take specialized

		Predicted Tags									
		NN	PN	VB	ADJ	ADV	AA	TA	Q	SC	Total
Gold Tags	NN	-	79	38	174	13	-	-	-	-	304
	PN	211	-	-	39	-	-	-	-	-	250
	VB	18	-	-	-	-	19	6	-	-	43
	ADJ	52	8	7	-	14	-	-	27	-	108
	ADV	29	1	5	31	-	-	-	-	19	85

Table 8: Confusion matrix of the tag pairs most confused by stand-alone tagger trained without BBC data.

morpheme that could distinguish them from common nouns (NN). This causes almost a half of PNs incorrectly tagged as NN. PN is usually marked as an adjective (ADJ) if it is used to refer to some property, state, or feature in the context (Ali et al., 2011), making tagger to confuse PN with ADJ.
Similarly, ADJ is marked with NN if head NN is dropped from the sentence. This causes a large number of NNs being tagged as ADJs (see Ali et al. (2011) for examples).
During the analysis of tagger output, we found several cases where we do not agree with the selection of part-of-speech assigned to certain words in the test data. Below we discuss such cases with examples.

Q as an ADJ: According to the Sajjad's tagset, words such as کئی, ہر are quantifiers. But in the test data, we see examples that mark a few quantifiers as ADJs. In Example 1, the quantifier کئی and in Example 2, the quantifier ہر are tagged as an ADJ.

(1) P|کے NN|پتنگوں SE|سے NN|سال ADJ|کئی PP|وہ
مقابلے|NN دیکھ|VB رہا|AA ہے|TA
woh kaī sāl se patangoñ ke muqāble dekh rahā he

(2) I|ہی NN|غفیر NN|جم P|کا NN|عوام NN|وقت ADJ|ہر
SM|۔ VB|دیکھا
her waqt ʾawām kā jme ğafer hī dīkhā .

NN as an ADJ: There are a few occurrences in which mostly the first or sometimes all nouns in compound noun are marked as ADJ. In Example 3, the first noun of the compound noun خریداری مراکز and in Example 4, all nouns of the compound noun اخبار فروش are marked as ADJ.

(3) ADJ|خریداری P|کے PN|پاسکو CC|اور NN|خوراک NN|محکمہ
AA|سکے VB|کر NEG|نہ NN|فروخت NN|گندم P|پر NN|مراکز
SM|۔
meḥakmah xorāk or pāsko ke xarīdārī marākəz per gandum farūxt nah kar sake .

(4) PN|سعید ADJ|فروش ADJ|اخبار ADJ|نوجوان PP|انہیں
PN|ناصر NN|ملازم P|کے NN|سروس CC|اور PN|علی
SM|۔ VB|پہنچا NN|دکھ ADJ|دلی P|پر NN|ہلاکت P|کی

inheñ nojawān axbār faroš saʾed ʾalī or koreīr sarwis ke mulāzim nāşir kī halākat per dilī dukh pohančā .

NN as PN: Although our tagger performs poorly on identifying proper nouns, especially the names of people and countries, we come across a few examples in the test data where NN are sometimes tagged as PN. In Example 5, کار and in Example 6, ملازم are tagged as PN whereas they are undoubtedly NN.

(5) AA|جا VB|کھینچی I|بھی PN|کار PN|سوزوکی SE|سے PP|اس
SM|۔ TA|ہے AA|سکتی
is se sūzūkī kār bhī khenčī jā saktī he .

(6) NN|پتنگوں SC|کہ VB|کہا P|نے PN|اقبال PN|ملازم PN|واپڈا
NN|بار Q|سینکڑوں P|کو PN|اتوار SE|سے NN|وجہ P|کی
NN|لوڈشیڈنگ VB|ہوتی SM|۔ TA|ہے
wāpḍā mulāzim iqbāl ne kahā keh patangoñ kī wajah se itwār ko synkaṛoñ bār loḍšyḍing hotī he .

PN as NN: We found a few cases where PN is tagged as NN. The frequency of these errors is still unknown and needs to be further investigated. Following are the examples where our tagger correctly predicts the PN but due to the incorrect gold tags, tagger's output for these nouns is considered wrong. On the contrary, there might be cases where our tagger benefits from the error made in gold tags.
In Example 7, the name of the stadium, نیشنل ہاکی سٹیڈیم, is tagged as NN whereas in Example 8, the name of railway sports board, ریلوے سپورٹس بورڈ is tagged as NN.

(7) ADJ|قومی ADJ|جاری P|میں NN|سٹیڈیم NN|ہاکی NN|نیشنل
P|میں NN|تربیت NN|کھلاڑی P|میں NN|کیمپ NN|ہاکی
ADJ|مصروف VB|ہیں SM|۔
našanal hākī satyḍym meñ jārī qomī hākī kemp meñ khilāṛī tarbyt meñ maşrof heñ .

(8) NN|ریلوے NN|منیجر NN|جنرل NN|صدارت P|کی REP|جس
CC|اور NN|صدر NN|ریلوے NN|سپورٹس NN|بورڈ NN|اقبال PN|
PN|صمد PN|خان P|نے VB|کی SM|۔
jis kī şadārət jenral manījar rylwe or şadar rylwe sporţs borḍ

	NN	PN	VB	ADJ	ADV
Before Error Corrections	87.62%	44.44%	94.64%	78.82%	41.13%
After Error Corrections	88.52%	46.86%	94.64%	84.09%	42.48%

Table 9: Taggers's individual accuracies of open class words before and after modifying the test data.

iqbāl şamad xān ne kī .

Other less frequent phenomena: The test data contains a few examples where ADJ is labeled as NN. In Example 9, سالہ as an NN in not entirely wrong as in other cases discusses above but in our opinion, ADJ is better choice for سالہ in this case.

(9) PN|ظفر NN|سالہ CA|15 WALA|والے VB|لوٹنے NN|پتنگ
PN|منٹو NN|ڈور AA|ہوئی VB|لوٹی SC|کہ VB|بتایا P|نے
SM|۔ TA|ہے VB|بکتی U|کلو NN|روپے CA|500 P|میں PN|پارک

patang lūṭne wāle 15 sāla ḍafar ne batāyā keh lūṭī hūī ḍor manṭo pārk meñ 500 rūpe kilo biktī he .

In a few other examples, the coordinating conjunct (CC) is tagged as subordinating conjunct (SC). In Example 10, coordinating conjunct یا is tagged as SC.

(10) P|میں NN|کچہری NN|ضلع SC|یا NN|تحصیل PP|وہ
SM|۔ VB|تھا NN|خواہشمند P|کا VB|بننے NN|چپڑاسی

woh teḥşyl yā ẓila' kačehrī meñ čapṛāsī banne kā xuwāhiš-mand thā .

We make a few error corrections in the test data and calculate the accuracy again on this modified test set. The modifications in the test data includes: assigning the correct tag (Q) to quantifiers when tagged as ADJ, ADV and CA; due to the difficulty of finding all such cases where NN or PN are incorrectly tagged as ADJ or NN simultaneously, we only make error corrections for those NN's and PN's that we found during our error analysis and mostly quoted in the examples above.
The accuracy we get after making these small changes in the test data is 88.74%. Table 9 shows individual accuracies of open class words on test data before and after making error corrections.

7. Conclusion

We release two useful resources for processing Urdu, a large monolingual corpus in plain text and automatically tagged and also a standalone tagger trained on the monolingual data. Hopefully, many NLP applications will benefit from these resources. Our main interest and follow-on work will be machine translation into Urdu.

8. Acknowledgments

We would like to thank Tafseer Ahmed for his valuable comments on tagger error analysis and verification of the errors contained in the test data.
This work has been using language resources developed and/or stored and/or distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2010013) and it is supported by the MosesCore project sponsored by the European Commission's Seventh Framework Programme (Grant Number 288487).

9. References

Ali, Aasim, Hussain, Sarmad, Malik, Kamran, and Siddiq, Shahid. (2011). Study of noun phrase in urdu. In *Conference on Language and Technology 2010 (CLT10)*, pages 44–54. Journal of Language and Literature Review, Vol. 1 No 1, 2011.

Baker, P., Hardie, A., McEnery, T., Cunningham, H., and Gaizauskas, R. (2002). Emille, a 67-million word corpus of indic languages: Data collection, mark-up and harmonisation. In *Proceedings of the 3rd Language Resources and Evaluation Conference (LREC'2002)*, pages 819–825. ELRA.

Giménez, J. and Márquez, L. (2004). Svmtool: A general pos tagger generator based on support vector machines. In *Proceedings of the 4th LREC*, Lisbon, Portugal.

Hardie, Andrew. (2003). Developing a tagset for automated part-of-speech tagging in urdu. Department of Linguistics, Lancaster University.

Humayoun, Muhammad. (2006). Urdu morphology, orthography and lexicon extraction. In *Master's Thesis*. Department of Computing Science, Chalmers University of Technology, oct.

Jawaid, Bushra and Bojar, Ondřej. (2012). Tagger voting for urdu. In *Proceedings of the Workshop on South and Southeast Asian Natural Language Processing (WSSANLP) at Coling 2012*, pages 135–144, Mumbai, India. Institute of Information Technologies (IIT) Bombay, Coling 2012 Organizing Committee.

Sajjad, Hassan and Schmid, Helmut. (2009). Tagging urdu text with parts of speech: a tagger comparison. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 692–700, Stroudsburg, PA, USA. Association for Computational Linguistics.