On Certain Aspects of Kazakh Part-of-Speech Tagging

Aibek Makazhanov, Zhandos Yessenbayev, Islam Sabyrgaliyev, and Anuar Sharafudinov
Nazarbayev University Research and Innovation System
53 Kabanbay batyr ave., Astana, Kazakhstan
E-mail: {aibek.makazhanov,zhyessenbayev,islam.sabyrgaliyev,anuar.sharaphudinov}@nu.edu.kz

Olzhas Makhambetov
National Information Technologies
8 Orynbor str., Astana, Kazakhstan
E-mail: olzhas.makhambetov@nitec.kz

***Abstract*—We compare and discuss various approaches to the problem of part of speech (POS) tagging of texts written in Kazakh, an agglutinative and highly inflectional Turkic language. In Kazakh a single root may produce hundreds of word forms, and it is difficult, if at all possible, to label enough training data to account for a vast set of all possible word forms in the language. Thus, current state of the art statistical POS taggers may not be as effective for Kazakh as for morphologically less complex languages, e.g. English. Also the choice of a POS tag set may influence the informativeness and the accuracy of tagging.**

I. Introduction

Kazakh is an agglutinative and highly inflectional Turkic language, spoken by more than 10 million people mainly in Kazakhstan, Russia, and China. Recent years have seen notable progress within research on Computational Processing of Kazakh Language. Namely, there have been studies on development of Kazakh corpora [1], [2], morphological analysis of Kazakh [2]–[6], spelling correction [3] and machine translation [7]. Unfortunately, the problem of POS tagging in Kazakh has not received as much research attention. There has been some work done by Gulila Altenbek et al., but the two papers that we found were written in Chinese and we could not study them due to lack of knowledge of the language.

As a prerequisite step for many advanced tasks of natural language processing (e.g. information extraction, machine translation, sentiment analysis, etc.) the POS tagging problem needs to be studied in detail. In the present preliminary study we seek to answer the following questions:

1) How does the choice of a tag set affect tagging accuracy?
2) Which of the existing methods better suits the task?
3) How much training data is required to achieve a decent accuracy?

To answer the first two questions we perform an experiment featuring two tag sets and two openly available statistical taggers. To answer the third question we conduct the experiment on three different train-test data samples.

The rest of the paper is organized as follows. In the following section we describe some of the existing approaches to the problem of POS tagging. In Section III we describe the process of Kazakh POS tag set design. Following that in Section IV we perform tagging experiments under various settings and provide an extensive discussion of the results. We draw conclusions and discuss the future work in Section V.

II. Related Work

There are numerous approaches to the problem of assigning POS tags to a sequence of words. Most of the tagging systems employ neural networks (NN), rules and statistical models. One of the pioneering works that applied NN was presented by Schmid [8]. The proposed method is based on the multilayer perceptron NN that is trained by backpropagation algorithm with momentum term and utilizes special lexicon consisting of the full form, suffix lexicon and default entry. In order to achieve good results this method may require large amounts of training data.

Among rule-based taggers we can mention the transformation-based learning (TBL) algorithm proposed by Brill [10]. TBL utilizes a greedy search approach, where at each iteration of learning: 1) it selects a rule that scores higher when applied to the recent state of the training set; 2) adds it to the ordered list of rules; 3) updates the training set using the selected rule. An initial assignment is made based on simple statistics. While this method seems attractive it may have some disadvantages: generated rules might not be optimal due to greedy learning; the method can be computationally infeasible. Another popular tagger (TreeTagger) presented by Schmid [11] utilizes decision trees (DTs) and a lexicon similar to the one used by Cutting et al. [9]. To estimate the transitional probabilities the author uses binary DTs, which are built recursively from trigrams in the training set. The efficiency of the method largely depends on the size of a training set.

The majority of taggers based on statistical methods utilize Markov models [12], [13]. One of the well-known works (TrigramsnTag - TnT) that can be easily adapted to different languages was authored by Brants [14]. TnT is an efficient

TABLE I
GRAMMATICAL ASPECTS CONSIDERED IN THE KLC TAG SET DESIGN

#	Grammatical Aspect	Code	Cardinality
1	Animacy	A	2
2	Number	N	2
3	Possessiveness	S	10
4	Person	P	8
5	Case	C	7
6	Negation	G	2
7	Tense	T	3
8	Mood	M	4
9	Voice	V	5

TABLE II
THE KLC KAZAKH POS TAG SET

#	POS	Tag	Paradigm	Capacity
1	impersonal noun	ZEP	ANSPC	314
2	personal noun	ZEQ	ANSPC	314
3	regular verb	ET	GTMVP	840
4	infinitive verb	ETU	GSC	196
5	auxiliary verb	ETK	P	8
6	auxiliary negative verb	ETB	P	8
7	auxiliary desiderative verb	KEL	GT	6
8	present participle	ESM	GNSPC	314
9	past participle	KSE	G	2
10	regular adjective	SE	P	8
11	comparative adjective	SES	P	8
12	superlative adjective	SEA	P	8
13	cardinal numeral	SN	NSPC	157
14	ordinal numeral	SNR	NSPC	157
15	collective numeral	SNZ	NSPC	157
16	fractional numeral	SNB	NSPC	157
17	regular adverb	US	-	1
18	comparative adverb	USS	-	1
19	superlative adverb	USA	-	1
20	personal pronoun	SIMZ	NSPC	229
21	demonstrative pronoun	SIMU	NSPC	157
22	interrogative pronoun	SIMS	NSPC	157
23	reflexive pronoun	SIMD	NSPC	157
24	indefinite pronoun	SIMB	NSPC	157
25	negative indefinite pronoun	SIMY	NSPC	157
26	universal indefinite pronoun	SIMP	NSPC	157
27	auxiliary nominal	KOM	C	7
28	preposition	SHS	-	1
29	conjunction	SHZ	-	1
30	particle	SHD	-	1
31	vocative interjection	OSP	-	1
32	vocative thought	OSQ	-	1
33	vocative emotion	OSO	-	1
34	onomatopoeia	ELK	-	1
35	modal word	MOD	-	1
36	foreign word	BOS	-	1
	total:			3844

data-driven tagger, which is based on second order Markov models. For probability smoothing the author uses linear interpolation of unigrams, bigrams and trigrams, estimating weights by deleted interpolations. Unknown words are handled by using word endings. In the work presented by Hakkani-Tur et al. [15] POS tagging is carried out by utilizing morphological disambiguation, where their trigram model achieves more than 90% accuracy (for Turkish). Recent advances include a tagger designed by Toutanova et al. [16]. In this work the authors use bidirectional dependency network with Maximum Entropy classifiers. The Viterbi algorithm is used to select the most likely sequence. The authors report 97.24% per-token accuracy on Penn Treebank Wall Street Journal corpus [17].

III. METHODOLOGY

In this section we describe[1] a positional POS tag set of Kazakh designed by Makhambetov et al. [1]. Hereinafter we will refer to the tag set as the KLC (Kazakh Language Corpus) tag set. When designing a POS tag set for agglutinative or/and fusional languages, a common practice is to employ a so called positional approach [18]–[20], representing a tag as a POS label accompanied by a *paradigm* string, whose positions denote certain grammatical aspects, say a verb mood, and accept certain values. The KLC tag set was designed following this principle.

Table I lists the grammatical aspects encoded by the tag set along with their codes and *cardinalities*, i.e. a number of values they accept. Table II provides a detailed description of the KLC tag set (excluding punctuation). For POS that accept inflectional suffixes the table lists respective paradigms together with *generative capacities*, i.e. the upper bound on a number of possible tags that can be generated from a given POS and the different combinations of the corresponding paradigms. The *maximum* size of the tag set equals to the total generative capacity, or 3844 tags. Depending on the level of granularity required for an application, some or even all grammatical aspects may be dropped or added back in, providing additional flexibility. Let us consider an example: *Mektepke bardym. - school.Dat go.Past.1sg - I went toschool.* Using the KLC tag set this sentence can be tagged as follows:

Mektepke/ZEP_A0N0S0P3C3 (ZEP - impersonal noun; A0 - inanimate; N0 - singular; S0 - no possessor; P3 - 3rd person; C3 - dative case) ***bardym***/ET_G0T3M1V0P1 (ET - regular verb; G0 - not negated; T3 - past tense; M1 - indicative mood; V0 - active voice; P1 - 1st person) ***./.***

IV. EXPERIMENTS

In this section we report and discuss the results of tagging experiments for two tag sets, two POS taggers and three data samples.

[1] Here we provide a rather brief description. For more details, please, consult the original work [1].

TABLE III
DATA SETS DESCRIPTION.

Data set	# word-tokens (train)	# sentences (train)	OOV rate
DATA100	534 563 (481 106)	46 800 (42 129)	0.08
DATA50	266 887 (240 198)	23 400 (21 063)	0.12
DATA20	106 524 (95 871)	9 360 (8 426)	0.18

A. Data Sets Description

Let us begin with a brief description of the data sets. For the experiments we used the annotated sub-corpus of the Kazakh Language Corpus [1]. As we want to assess the impact of the size of training data on the accuracy of tagging, we organize the data into three data sets: (i) the biggest set contains the data as a whole, and we denote this set as DATA100 (100%); (ii) the second set is referred to as DATA50, and it is twice as small; (iii) finally, the third set contains only 20% (DATA20) of the entire labeled data. We further split each of the three sets into a 90%-training 10%-testing samples. Table III shows the basic characteristics of the data sets in terms of quantities of word-tokens (punctuation does not count) and sentences, as well as Out Of Vocabulary (OOV) tokens rate, i.e. a portion of tokens in a test set that was not observed during the training. As it can be seen, the entire data consists of more than 530K words and almost 47K sentences. The sizes of training sets (both in words and sentences) are given in parenthesis. Also, for a 90-10 data split the OOV rates are fairly high, and they grow as the data sets become smaller.

B. Experimental Setup

The KLC annotation scheme allows us to fine tune the level of grammatical complexity of a given set of POS labels. For a preliminary study we took "all or nothing" approach and compared a tag set that incorporates all of the grammatical aspects (except *animacy*[2]) to a tag set that consists only of the 36 basic tags.

To compare different methods of tagging, we resort to a comparison of two statistical taggers, namely the Stanford bidirectional maximum entropy tagger [21] and the HMM-based Tree tagger [11] that also employs decision trees. We chose these taggers mainly due to their reported high-accuracy, availability, and ease of use. However, we certainly plan to experiment with open source implementations of rule-based and classification-based taggers [22].

Tables IV and V show the results of the experiment with the [NCPGMVT][3] and [] tag sets, i.e. the complete and the basic KLC tag sets. While the former tag set contains a total of 3844 tags, only 1049 (without punctuation) were found in the training set. The latter tag set consists of the 36 basic tags. We report the results for each tagger-data set pair, e.g. SF-D100 denotes the Stanford tagger trained on the complete data set, similarly TT-D50 means that the TreeTagger was trained on the smaller data set. The evaluation is carried out in terms of the per-token, per-sentence, and OOV accuracies. Unfortunately, as much as we tried the Stanford tagger could not be trained on the data labeled with the [NCPGMVT] tag set, presumably due to a large number of tags.

TABLE IV
POS TAGGING USING A [NCPGMVT] TAG SET.

tagger-data	per-tok. acc.	per-sent. acc.	OOV acc.
SF-D100	—	—	—
SF-D50	—	—	—
SF-D20	—	—	—
TT-D100	0.83	0.16	0.18
TT-D50	0.80	0.12	0.19
TT-D20	0.75	0.07	0.18

TABLE V
POS TAGGING USING A [] TAG SET.

tagger-data	per-tok. acc.	per-sent. acc.	OOV acc.
SF-D100	0.90	0.32	0.57
SF-D50	0.89	0.29	0.58
SF-D20	0.86	0.22	0.55
TT-D100	0.88	0.29	0.56
TT-D50	0.87	0.26	0.56
TT-D20	0.84	0.19	0.56

As it can be seen, the choice of a tag set does influence the accuracy of tagging quite a bit. On a smaller, much concise tag set, TT gained performance in all of the three metrics (cf., Table V), most notably in the OOV accuracy. While it seems like a very logical conclusion that reducing the size of a tag set improves the accuracy, one should not forget about a possible information loss caused by such a reduction. For instance in Kazakh, case suffixes bear prepositional meanings, thus tagging a nominal with a POS label that also provides an accurate case information is quite beneficial, e.g. for a consequent parsing. Moreover, previous works on the subject studied a much greater number of configurations of positional tag sets, and concluded that increasing the size of a tag set does not necessarily decrease the tagging accuracy [23] and that for agglutinative languages omitting grammatical aspects may hurt the accuracy of *n*-gram tagging [24]. Thus, to draw a final conclusion on the problem of the choice of a tag set for Kazakh, more experiments need to be conducted with various tag set configurations.

As for the choice of the POS taggers, although the Stanford tagger is slightly more accurate than TT when used with the basic tag set, it fails to work properly on the complete tag set. Of course, it is a technicality, and probably our own fault, but even for the basic tag set it took SF much longer to train and to run on the test data. Given that in many applications that use POS tagging speed is an issue, TT seems preferable. Overall, these two taggers represent the same branch of statistical taggers, and as we mentioned

[2] We skip the animacy because this aspect is not expressed morphologically.

[3] Each capital letter in the abbreviation (NCPGMVT) indicates that a correspondingly encoded grammatical aspect (cf., Table I) is incorporated in the tag set.

earlier we plan to test open source implementations of rule based and classifier based taggers using the NLTK toolkit [22]. Even within the statistical approaches there are methods better suited for agglutinative languages, that typically utilize formal methods of morphological analysis [25] or heuristics based on leveraging plain text word endings [26]. These method we also consider to implement.

Finally, we would like to discuss the training data amount issue. With a gradual reduction of a training set size, we observe 4% decrease in per-token accuracy, and almost 10% decrease in per-sentence accuracy for both taggers. The OOV accuracy turned out to be less prone to a decrease in the size of training data, which is expectable given that OOV tokens do not appear during training. While it is hard to estimate exactly how much training data is enough, we can speculate that if an 80% decrease in the amount of data decreases the accuracy by 4%, to get a per-token accuracy of 97-98%[4], we need at least 1.5 times more data than we currently posses.

V. Conclusions and Future Work

We have discussed a number of issues related to the POS tagging for Kazakh language. By performing a series of experiments we tried to assess the role of tag sets, taggers and the amount of training data in the development of a high quality and accurate approaches. We concluded that certain POS tag sets better suit the task, and that designing an optimal tag set still remains an open problem. Same goes for choosing an effective tagger and acquiring more training data. In our future work we plan to address these problems.

References

[1] O. Makhambetov, A. Makazhanov, Z. Yessenbayev, B. Matkarimov, I. Sabyrgaliyev, and A. Sharafudinov, "Assembling the kazakh language corpus," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, October 2013, pp. 1022–1031. [Online]. Available: http://www.aclweb.org/anthology/D13-1104

[2] G. Altenbek and W. Xiao-long, "Kazakh segmentation system of inflectional affixes," in *Joint Conference on Chinese Language Processing*. CIPS-SIGHAN, 2010, pp. 183–190.

[3] A. Makazhanov, O. Makhambetov, I. Sabyrgaliyev, and Z. Yessenbayev, "Spelling correction for kazakh," in *Proceedings of the 2014 Computational Linguistics and Intelligent Text Processing*. Kathmandu, Nepal: Springer Berlin Heidelberg, 2014, pp. 533–541.

[4] A. Sharipbayev, G. Bekmanova, B. Ergesh, A. Buribayeva, and M. K. Karabalayeva, "Intellectual morphological analyzer based on semantic networks," in *Proceedings of the OSTIS-2012*, 2012, pp. 397–400.

[5] G. Kessikbayeva and I. Cicekli, "Rule based morphological analyzer of kazakh language," in *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 46–54. [Online]. Available: http://www.aclweb.org/anthology/W/W14/W14-2806

[6] H. R. Zafer, B. Tilki, A. Kurt, and M. Kara, "Two-level description of kazakh morphology," in *Proceedings of the 1st International Conference on Foreign Language Teaching and Applied Linguistics (FLTAL11)*, Sarajevo, May 2011.

[7] A. S. Assem Shormakova, Aida Sundetova, "Features of machine translation of different systemic languages using an apertium platform (with an example of english and kazakh languages)," *International Journal of Soft Computing and Software Engineering [JSCSE]*, pp. 255–259, 2013, 10.7321/jscse.v3.n3.38. [Online]. Available: http://dx.doi.org/10.7321/jscse.v3.n3.38

[8] H. Schmid, "Part-of-speech tagging with neural networks," in *Proceedings of the 15th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1994, pp. 172–176.

[9] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, "A practical part-of-speech tagger," in *Proceedings of the third conference on Applied natural language processing*. Association for Computational Linguistics, 1992, pp. 133–140.

[10] E. Brill, "A simple rule-based part of speech tagger," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 112–116.

[11] H. Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of international conference on new methods in language processing*, vol. 12. Manchester, UK, 1994, pp. 44–49.

[12] R. Garside, "The CLAWS word-tagging system," in *The Computational Analysis of English: a corpus-based approach*, R. Garside, G. Leech, and G. Sampson, Eds. London: Longman, 1987, pp. 30–41.

[13] S. M. Thede and M. P. Harper, "A second-order hidden markov model for part-of-speech tagging," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 1999, pp. 175–182.

[14] T. Brants, "Tnt: a statistical part-of-speech tagger," in *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 2000, pp. 224–231.

[15] D. Z. Hakkani-Tür, K. Oflazer, and G. Tür, "Statistical morphological disambiguation for agglutinative languages," *Computers and the Humanities*, vol. 36, no. 4, pp. 381–410, 2002.

[16] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003, pp. 173–180.

[17] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: the penn treebank," *Comput. Linguist.*, vol. 19, no. 2, pp. 313–330, Jun. 1993. [Online]. Available: http://dl.acm.org/citation.cfm?id=972470.972475

[18] K. Oflazer, B. Say, D. Z. Hakkani-Tür, and G. Tür, "Building a turkish treebank," in *Treebanks*. Springer, 2003, pp. 261–277.

[19] J. Hajič and B. Hladká, "Tagging inflective languages: prediction of morphological categories for a rich, structured tagset," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ser. ACL '98. Stroudsburg, PA, USA: Association for Computational Linguistics, 1998, pp. 483–490. [Online]. Available: http://dx.doi.org/10.3115/980845.980927

[20] J. Hana and A. Feldman, "A positional tagset for russian," *Proceedings of LREC-10. Malta*, 2010.

[21] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, ser. EMNLP '00. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 63–70. [Online]. Available: http://dx.doi.org/10.3115/1117794.1117802

[22] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[23] D. Elworthy, "Tagset design and inflected languages," in *In EACL SIGDAT workshop iFrom Texts to Tags: Issues in Multilingual Language Analysis*, 1995, pp. 1–10.

[24] A. Feldman, "Tagset design, inflected languages, and n-gram tagging," *Editors: Paul Robertson and John Adamson*, vol. 3, no. 1, p. 151, 2008.

[25] H. Sak, T. Güngör, and M. Saraçlar, "A stochastic finite-state morphological parser for turkish," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ser. ACLShort '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 273–276. [Online]. Available: http://dl.acm.org/citation.cfm?id=1667583.1667667

[26] B. T. Dinçer, B. Karaoglan, and T. Kisla, "A suffix based part-of-speech tagger for turkish," in *ITNG*, 2008, pp. 680–685.

[4] The current state of the art for many languages, including a closely related Turkish.