A free/open-source hybrid morphological disambiguation tool for Kazakh

Conference Paper · April 2016

DOI: 10.13140/RG.2.2.12467.43045

8 authors, including:

Zhenisbek Assylbekov
Nazarbayev University
9 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Francis Tyers
University of Alicante
21 PUBLICATIONS **188** CITATIONS
SEE PROFILE

A free/open-source hybrid morphological disambiguation tool for Kazakh

Zhenisbek Assylbekov*, Jonathan North Washington†, Francis Tyers‡, Assulan Nurkas*,
Aida Sundetova§, Aidana Karibayeva§, Balzhan Abduali§, Dina Amirova§

*School of Science and Technology, Nazarbayev University
†Departments of Linguistics and Central Eurasian Studies, Indiana University
‡HSL-fakultehta, UiT Norgga árktalaš universitehta
§Information Systems Department, Al-Farabi Kazakh National University

***Abstract*—This paper presents the results of developing a morphological disambiguation tool for Kazakh. Starting with a previously developed rule-based approach, we tried to cope with the complex morphology of Kazakh by breaking up lexical forms across their derivational boundaries into inflectional groups and modeling their behavior with statistical methods. A hybrid rule-based/statistical approach appears to benefit morphological disambiguation demonstrating a per-token accuracy of 91% in running text.**

I. Introduction

In this paper, we present a free/open-source hybrid morphological disambiguation tool for Kazakh. Morphological disambiguation is the task of selecting the sequence of morphological parses corresponding to a sequence of words, from the set of possible parses for those words. Morphological disambiguation is an important step for a number of NLP tasks and this importance becomes more crucial for agglutinative languages such as Kazakh, Turkish, Finnish, Hungarian, etc. For example, by using a morphological analyzer together with a disambiguator the perplexity of a Turkish language model can be reduced significantly [1]. Kazakh (as well as any morphologically rich language) presents an interesting problem for statistical natural language processing since the number of possible morphological parses is very large due to the productive derivational morphology [2, 3]. In this work we combine rule-based [4] and statistical [5] approaches to disambiguate a Kazakh text: the output of a morphological analyzer is pre-processed using constraint-grammar rules [6], and then the most probable sequence of analyses is selected. Our combined approach works well even with a small hand-annotated training corpus. The performance of the presented hybrid system can likely be improved further when a larger hand-tagged corpus becomes available.

In Section II, we present relevant properties of Kazakh. Then, in Section III, we review the related work on part-of-speech (POS) tagging and morphological disambiguation. In Section IV, we describe the statistical model for morphological disambiguation. We finally present and discuss our results in Section V.

II. Kazakh

Kazakh (natively қазақ тілі, қазақша) is a Turkic language belonging to the Kypchak (or Qıpçaq) branch, closely related to Nogay (or Noğay) and Qaraqalpaq. It is spoken by around 13 million people in Kazakhstan, China, Mongolia, and adjacent areas [7].

Kazakh is an agglutinative language, which means that words are formed by joining suffixes to the stem. A Kazakh word can thus correspond to English phrases of various length as shown below:

дос	friend
дос**тар**	friend**s**
достар**ым**	**my** friends
достар**ымыз**	**our** friends
достарымыз**да**	**at** our friends
достарымызда**мыз**	**we are** at our friends

The effect of rich morphology can be observed in parallel Kazakh-English texts. Table below provides the vocabulary sizes, type-token ratios (TTR) and out-of-vocabulary (OOV) rates of Kazakh and English sides of a parallel corpus used in [8].

	English	Kazakh
Vocabulary size	18,170	35,984
Type-token ratio	3.8%	9.8%
OOV rate	1.9%	5.0%

It is easy to see that rich morphology leads to sparse data problems for statistical natural language processing of Kazakh, be it tasks in machine translation, text categorization, sentiment analysis, etc. A common approach (see [9, 10, 11, 12]) applied for morphologically rich languages is to convert surface forms into lexical forms (i.e. analyze words), and then perform some morphological segmentation for the lexical forms (i.e. split analyzes). The segmentation schemes are usually motivated by linguistics and the domain of intended use. For example, for a Kazakh-English word alignment task we could be in

favor of the following segmentation of the above mentioned word *достарымыздамыз*[1]

достар	ымыз	да	мыз	
дос⟨n⟩⟨pl⟩	⟨px1pl⟩	⟨loc⟩	+e⟨cop⟩	⟨p1⟩⟨pl⟩
friends	our	at	are	we

since each segment of the Kazakh word would then correspond to a single word in English. The problem is that often for a word in Kazakh we have more than one way to analyze it, as in the example below:

‘in 2009 , we started the construction works .’
2009 жылы біз құрылысты бастадық .

	жылы⟨adj⟩	‘warm’
	жылы⟨adj⟩⟨advl⟩	‘warmly’
→	**жыл⟨n⟩⟨px3sp⟩⟨nom⟩**	**‘year’**
	жылы⟨adj⟩⟨subst⟩⟨nom⟩	‘warmth’

Selecting the correct analysis from among all possible analyses is called morphological disambiguation. Due to productive derivational morphology this task itself suffers from data sparseness. To alleviate the data sparseness problem we break down the full analyses into smaller units – inflectional groups. An inflectional group is a tag sequence split by a derivation boundary. For example, in the sentence that follows, the word *айналасындағыларға* ‘to the ones in his vicinity’ is split into root r and two inflectional groups, g_1 and g_2, the first containing the tags before the derivation boundary *-ғы* and the second containing the derivation boundary and subsequent tags.

Жаңгір хан мен оның **айналасындағыларға** . . .

(айнала)·(сын·да)·(ғы·лар·ға)

$\underbrace{(\text{айнала})}_{r}\cdot\underbrace{(\text{n·px3sp·loc})}_{g_1}\cdot\underbrace{(\text{subst·pl·dat})}_{g_2}$

We will heavily exploit the following observation of dependency relationships which was made by Hakkani-Tür et al. [5, p. 387] for Turkish, but is valid for Kazakh as well: When a word is considered to be a sequence of inflectional groups, syntactic relation links only emanate from the *last inflectional group* of a (dependent) word, and land on *one of the inflectional groups* of the (head) word on the right.

III. Related work

Morphological disambiguation of inflectional and agglutinative languages was inspired by part-of-speech (POS) tagging techniques. Due to Chomsky’s criticism of the inadequacies of Markov models [14, ch. 3], the lack of training data and computing resources to pursue an ‘empirical’ approach to natural language, early work on POS tagging using Markov chains had been largely abandoned by the early sixties. The earliest ‘taggers’ were simply programs that looked up the category of words in a dictionary. The first well-known program which attempted to assign tags based on syntagmatic contexts was the rule-based program presented in [15], though roughly the same idea is present in [16]. One of the most well-known corpora, Brown corpus, was automatically pre-tagged with a rule-based tagger, TAGGIT [17]. The earliest probabilistic tagger known to us is [18]. One of the first Markov Model taggers was created at the University of Lancaster as part of Lancaster-Oslo-Bergen corpus tagging effort [19, 20]. The type of Markov Model tagger that tags based on both word probabilities and tag transition probabilities was introduced by Church [21] and DeRose [22]. All these taggers are trained on hand-tagged data. Kupiec [23], Cutting et al. [24], and others show that it is also possible to train a Hidden Markov Model (HMM) tagger on unlabeled data, using the EM algorithm [25]. An experiment by Merialdo [26], however, indicates that with even a small amount of training data, a tagger trained on hand-tagged data worked better than one trained via EM. Other notable approaches in POS tagging are Brill’s transformation-based learning paradigm [27], the memory-based tagging paradigm [28], and the maximum entropy-based approach [29].

Morphological disambiguation in inflectional or agglutinative languages with complex morphology involves determining not only the major or minor parts-of-speech, but also *all* relevant lexical and morphological features of surface forms. Levinger et al. [30] suggested an approach for morphological disambiguation of Hebrew. Hajič and Hladká [31] have used maximum entropy modeling approach for morphological disambiguation of Czech, an inflectional language. Hajič [32] extended this work to 5 other languages including English and Hungarian (an agglutinative language). Ezeiza et al. [33] have combined stochastic and rule-based disambiguation methods for Basque, which is also an agglutinative language. Megyesi [34] has adapted Brill’s POS tagger with extended lexical templates to Hungarian.

From all languages which are widely researched nowadays Turkish is the closest one to Kazakh. Previous approaches to morphological disambiguation of Turkish text had employed constraint-based methods (Oflazer and Kuruöz [35]; Oflazer and Tür [36, 37]), statistical methods (Hakkani-Tür et al. [5], Sak et al. [38]), or both (Yuret and Türe [39], Kutlu and Cicekli [40]).

Recently, some work has been done towards developing morphological disambiguation tools for Kazakh. Salimzyanov et al. [4] provide constraint grammar rules which reduce ambiguity from 2.4 to 1.4 analyzes per form in a running text. Makhambetov et al. [41] present a comparison of part-of-speech taggers trained on the Kazakh National Corpus [42]: the best result obtained, using the full training data of around 600,000 tokens was a per-token accuracy of 86% when cross-validated on the same training data with 10 folds. Kessikbayeva and Cicekli [43] present a transformation-based morphological disambiguator for Kazakh which is trained on hand-annotated corpus of over 30,000 words and gains 87% accuracy when tested against a test data of around 15,000 words.

[1]hereinafter we use the Apertium tagset [13] for analyzed forms

IV. Statistical morphological disambiguation

Following [44], we will use the notation in Table I. We use

w_i	the word (token) at position i in the corpus
t_i	the tag of w_i
$w_{i,i+m}$	the words occurring at positions i through $i+m$
$t_{i,i+m}$	the tags $t_i \cdots t_{i+m}$ for $w_i \cdots w_{i+m}$
r_i	the root of w_i
$g_{i,k}$	the k-th inflectional group of w_i
n	length of a text chunk (be it a sentence, a paragraph or a whole text)
$\mathbf{w}$	the words $w_{1,n}$ of a text chunk
$\mathbf{t}$	the tags $t_{1,n}$ for $w_{1,n}$

TABLE I: 'Notation'

subscripts to refer to words and tags in particular positions of the sentences and corpora we tag. We use superscripts to refer to word types in the lexicon of words and to refer to tag types in the tag set.

The basic mathematical object with which we deal here is the joint probability distribution $\Pr(\mathbf{W} = \mathbf{w}, \mathbf{T} = \mathbf{t})$, where the random variables $\mathbf{W}$ and $\mathbf{T}$ are a sequence or words and a sequence of tags. We also consider various marginal and conditional probability distributions that can be constructed from $\Pr(\mathbf{W} = \mathbf{w}, \mathbf{T} = \mathbf{t})$, especially the distribution $\Pr(\mathbf{T} = \mathbf{t})$. We generally follow the common convention of using uppercase letters to denote random variables and the corresponding lowercase letters to denote specific values that the random variables may take. When there is no possibility for confusion, we write $\Pr(\mathbf{w}, \mathbf{t})$, and use similar shorthands throughout.

In this compact notation, morphological disambiguation is the problem of selecting the sequence of morphological parses (including the root), $\mathbf{t} = t_1 t_2 \cdots t_n$, corresponding to a sequence of words $\mathbf{w} = w_1 w_2 \cdots w_n$, from the set of possible parses for these words:

$$\arg\max_{\mathbf{t}} \Pr(\mathbf{t}|\mathbf{w}). \tag{1}$$

Using Bayes' rule and taking into account that $\mathbf{w}$ is constant for all possible values $\mathbf{t}$, we can rewrite (1) as:

$$\arg\max_{\mathbf{t}} \frac{\Pr(\mathbf{t}) \times \Pr(\mathbf{w}|\mathbf{t})}{\Pr(\mathbf{w})} = \arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \times \Pr(\mathbf{w}|\mathbf{t}) \tag{2}$$

In Kazakh, given a morphological analysis[2] including the root, there is only one surface form that can correspond to it, that is, there is no morphological generation ambiguity. Therefore,

$$\Pr(\mathbf{w}|\mathbf{t}) = 1,$$

and the morphological disambiguation problem (2) is simplified to finding the most probable sequence of parses:

$$\arg\max_{\mathbf{t}} \Pr(\mathbf{t}) \tag{3}$$

Keep in mind that the search space in equations (1)–(3) is not equal to the set of all hypothetically possible sequences $\mathbf{t}$. Instead it is limited to only the set of parse sequences that can correspond to $\mathbf{w}$. Such limited set is obtained as a full or constrained output of a morphological analysis tool.

[2]We use the terms morphological analysis or parse interchangeably, to refer to individual distinct morphological parses of a token.

A. Derivation

Using the chain rule, the probability in (3) can always be rewritten as:

$$\Pr(\mathbf{t}) = \prod_{i=1}^{n} \Pr(t_i|t_{1,i-1}). \tag{4}$$

It is important to realize that equation (4) is not an approximation. We are simply asserting in this equation that when we generate a sequence of parses, we can firstly choose the first analysis. Then we can choose the second parse given our knowledge of the first parse. Then we can select the third analysis given our knowledge of the first two parses, and so on. As we step through the sequence, at each point we make our next choice given our complete knowledge of the all our previous choices.

The conditional probabilities on the right-hand side of equation (4) cannot all be taken as independent parameters because there are too many of them. In the bigram model, we assume that

$$\Pr(t_i|t_{1,i-1}) \approx \Pr(t_i|t_{i-1}).$$

That is, we assume that the current analysis is only dependent on the previous one. With this assumption we get the following:

$$\Pr(\mathbf{t}) \approx \prod_{i=1}^{n} \Pr(t_i|t_{i-1}). \tag{5}$$

However, the probabilities on the right-hand side of this equation still cannot be taken as parameters, since the number of possible analyzes is very large in morphologically rich languages. Following the discussion from Section II we split morphological parses across their derivational boundaries, i.e. we consider morphological analysis as a sequence of root (r_i) and inflectional groups ($g_{i,k}$), and therefore, each parse t_i can be represented as $(r_i, g_{i,1}, \ldots, g_{i,n_i})$. Then the probabilities $\Pr(t_i|t_{i-1})$ can be rewritten as:

$$\begin{aligned}
&\Pr(t_i|t_{i-1}) \\
&= \Pr((r_i, g_{i,1}, \ldots, g_{i,n_i})|(r_{i-1}, g_{i-1,1}, \ldots, g_{i-1,n_{i-1}})) \\
&= \{\text{chain rule}\} = \Pr(r_i|(r_{i-1}, g_{i-1,1}, \ldots, g_{i-1,n_{i-1}})) \\
&\times \Pr(g_{i,1}|(r_{i-1}, g_{i-1,1}, \ldots, g_{i-1,n_{i-1}}), r_i) \times \ldots \times \\
&\times \Pr(g_{i,n_i}|(r_{i-1}, g_{i-1,1}, \ldots, g_{i-1,n_{i-1}}), r_i, g_{i,1}, \ldots, g_{i,n_i-1})
\end{aligned} \tag{6}$$

In order to simplify this representation we throw in the following independence assumptions

$$\Pr(r_i|(r_{i-1}, g_{i-1,1}, \ldots, g_{i-1,n_{i-1}})) \approx \Pr(r_i|r_{i-1}), \tag{7}$$

$$\begin{aligned}
&\Pr(g_{i,k}|(r_{i-1}, g_{i-1,1}, \ldots, g_{i-1,n_{i-1}}), r_i, g_{i,1}, \ldots, g_{i,k-1}) \\
&\qquad \approx \Pr(g_{i,k}|g_{i-1,n_{i-1}}),
\end{aligned} \tag{8}$$

i.e. we assume that the root in the current parse depends only on the root of the previous parse, and each inflectional group in the current parse depends only on the last inflectional group of the previous parse (this last assumption is motivated by the

remark at the end of Section II). Now, from (6), (7), and (8) we get:

$$\Pr(t_i|t_{i-1}) \approx \underbrace{\Pr(r_i|r_{i-1}) \prod_{k=1}^{n_i} \Pr(g_{i,k}|g_{i-1,n_{i-1}})}_{\Pr_b(t_i|t_{i-1})}, \quad (9)$$

where we define $r_0 =$ `'.'` and $g_{0,n_0} =$ `'<sent>'`. Now putting together (5) and (9) we have:

$$\Pr(\mathbf{t}) \approx \prod_{i=1}^{n} \Pr(t_i|t_{i-1}) \approx \underbrace{\prod_{i=1}^{n} \left[\Pr(r_i|r_{i-1}) \prod_{k=1}^{n_i} \Pr(g_{i,k}|g_{i-1,n_{i-1}}) \right]}_{\Pr_b(\mathbf{t})}. \quad (10)$$

$\Pr(r^l|r^m)$ and $\Pr(g^l|g^m)$ are parameters (root and IG probabilities) which can be estimated using manually disambiguated texts.

B. Parameters estimation

Assume we are observing a sequence of n tokens w_1, w_2, …, w_n, and each token was manually disambiguated, i.e. we posses a sequence of corresponding parses t_1, t_2, …, t_n. Then the likelihood for our data is given by the equation (10), and in order to find maximum likelihood estimates for the parameters $\Pr(r^l|r^m)$ and $\Pr(g^l|g^m)$ we need to solve the following optimization problem:

$$\prod_{i=1}^{n} \left[\Pr(r_i|r_{i-1}) \prod_{k=1}^{n_i} \Pr(g_{i,k}|g_{i-1,n_{i-1}}) \right] \longrightarrow \max \quad (11)$$

$$\sum_l \Pr(r^l|r^m) = 1, \qquad \sum_l \Pr(g^l|g^m) = 1. \quad (12)$$

Using the method of Lagrange multipliers [45] one can show that the solution of (11) subject to constraints (12) is given by:

$$\Pr_{\text{MLE}}(r^l|r^m) = \frac{C(r^m, r^l)}{C(r^m)}, \ \Pr_{\text{MLE}}(g^l|g^m) = \frac{C(g^m, g^l)}{C(g^m)}, \quad (13)$$

where $C(r^m)$ is the number of occurrences of r^m, $C(r^m, r^l)$ is the number of occurrences of r^m followed by r^l, $C(g^m)$ is the number of occurrences of g^m, $C(g^m, g^l)$ is the number of parses with g^m as the last IG followed by a parse containing g^l. However, the maximum likelihood estimates suffer from the following problem: What if a bigram has not been seen in training, but then shows up in the test data? Using the formulas (13) we would assign unseen bigrams a probability of 0. Such approach is not very useful in practice. If we want to compare different possible parses for a sentence, and all of them contain unseen bigrams, then each of these parses receives a model estimate of 0, and we have nothing interesting to say about their relative quality. Since we do not want to give any sequence of words zero probability, we need to assign some probability to unseen bigrams. Methods for adjusting the empirical counts that we observe in the training corpus to the expected counts of n-grams in previously unseen text involve smoothing, interpolation and back-off: they have been discussed by Good [46], Gale and Sampson [47], Written and Bell [48], Knesser and Ney [49], Chen and Goodman [50]. The latter paper presents an extensive empirical comparison of several of widely-used smoothing techniques and introduces a variation of Kneser–Ney smoothing that consistently outperforms all other algorithms evaluated. We used it for estimating the parameters of the bigram model (10).

C. Tagging with the Viterbi algorithm

Once parameters are estimated we could evaluate the bigram model (10) for all possible parses $t_{1,n}$ of a sentence of length n, but that would make tagging exponential in the length of the input that is to be tagged. An efficient tagging algorithm is the Viterbi algorithm (Algorithm 1). It has three steps:

Algorithm 1 Algorithm for tagging

Require: a sentence $w_{1,n}$ of length n
Ensure: a sequence of analyzes $t_{1,n}$
1: $\delta_0($`('.', <sent>`$)) = 1.0$
2: $\delta_0(t) = 0.0$ for $t \neq$ `('.', <sent>`
3: **for** $i = 1$ **to** n **step** 1 **do**
4: **for** all candidate parses t^j **do**
5: $\delta_i(t^j) = \max_{t^k}[\delta_{i-1}(t^k) \times \Pr_b(t^j|t^k)]$
6: $\psi_i(t^j) = \arg\max_{t^k}[\delta_{i-1}(t^k) \times \Pr_b(t^j|t^k)]$
7: **end for**
8: **end for**
9: $X_n = \arg\max_{t^j} \delta_n(t^j)$
10: **for** $j = n - 1$ **to** 1 **step** -1 **do**
11: $X_j = \psi_{j+1}(X_{j+1})$
12: **end for**

initialization (lines 1–2), induction (lines 3–8), termination and path readout (lines 9–12). We compute two functions $\delta_i(t^j)$, which gives us the probability of parse t^j for word w_i, and $\psi_{i+1}(t^j)$, which gives us the most likely parse at word w_i given that we have the parse t^j at word w_{i+1}. A more detailed discussion of the Viterbi algorithm for tagging is provided in [51].

V. Experiments and results

A. Training and test data

We selected thirteen most viewed articles from Kazakh Wikipedia according to 2014 page counts data (see Table II), and used all of them except ‘Басты бет’, ‘CERN’, and ‘Жапония префектуралары’ to create a training set[3]. This totaled in approximately 12.5K words (15.7K tokens). We performed morphological analysis for our texts using an open-source finite-state morphological transducer *apertium-kaz* [52]. It is based on Helsinki Finite-State Toolkit and is

[3] ‘Басты бет’ is not an article, it is a main page of Kazakh Wikipedia; articles ‘CERN’ and ‘Жапония префектуралы’ do not contain much text

Article title	Views	Tokens
Басты бет	1,674,069	–
Жапония	877,693	3,211
Біріккен Ұлттар Ұйымы	807,058	793
CERN	648,464	–
Иран	602,001	2,879
Жапония префектуралары	551,394	–
Футболдан әлем чемпионаты 2014	333,988	257
Жапония Ұлттық футбол құрама командасы	321,249	146
Eurovision ән конкурсы 2010	312,183	101
Абай Құнанбайұлы	242,151	4,083
Радиан	187,225	39
Жасуша	145,010	1,789
Шоқан Шыңғысұлы Уәлиханов	119,780	2,408
		15,706

TABLE II: Most viewed articles of Kazakh Wikipedia in 2014

available within the Apertium project [13]. The analysis was carried out by calling `lt-proc` command of the Lttoolbox [53]. A preliminary disambiguation was performed through Constrained Grammar rules [6] by calling the `cg-proc` command, which decreased ambiguity from 2.4 to 1.4 analyses per form on average. The remaining disambiguation was done manually in the following way: the texts were disambiguated independently by two different annotators. Unfortunately, spot-checking annotations showed that they were rather noisy: this was mainly due to the lack of annotation guidelines. Most common mistakes were connected with:

- choosing between `<attr>` (attributive) and `<nom>` (nominative) in noun-noun compounds: e.g. in *көрші елдер* 'neighbouring countries' the word *көрші* 'neighbour' should be tagged as `<n><attr>` (attributive noun), but in *әлем чемпионаты* 'world championship' the word *әлем* 'world' should be tagged as `<n><nom>` (noun in nominative case);
- choosing between `<cnjcoo>` (conjunction) and `<postadv>` (postadverb) for the words да/де/та/те: e.g. in *Үстелде қалам да, қарындаш та, дәптер де жатыр* 'There are pen, pencil and notebook on the table' they should be tagged as `<cnjcoo>`, but in *Мен де барамын* 'I will also go' it should be tagged as `<postadv>`;
- choosing between `<det><dem>` (demonstrative determiner) and `<prn>` (pronoun) for the words *бұл, мынау, осы, мына, анау, ана, сол* 'this, that': e.g. in *Мынау үй жаңа* 'This house is new' the word *мынау* should be tagged as `<det><dem>`, but in *Мынау – терезе емес* 'This is not a window' the word *мынау* should be tagged as `<prn>`;
- choosing between `<ger>` (gerund) and `<n>` (noun) for verbs in a dictionary form: e.g. in *Кітап оқу адамдарды ақылдырақ етеді* 'Reading books makes people wiser' the word *оқу* 'to read' should be tagged as `<ger>`, but in *Оқу басталды* 'Classes began' the word *оқу* 'study' should be tagged as `<n>`.

Based on these and other types of annotation mistakes we developed a set of guidelines[4], asked annotators to resolve the differences in annotations and fix them where necessary using the mentioned guidelines.

In order to enrich our model with more roots we extracted unambiguous sequences of 1,509,480 tokens in a corpus of 2,128,642 tokens and used these unambiguous sequences in addition to hand-annotated texts from Table II for estimating root probabilities.

For our test data we selected several texts from the free/open-source Kazakh treebank [54], which is based on universal dependency (UD) annotation standards. These texts are morphologically disambiguated and annotated manually for dependency structure, but for our purposes we used only morphological annotations. We made sure that the document 'wikipedia' does not overlap with our training data. Composition of the test data is given below:

Document	Description	Tokens
Шымкент	Wikipedia article (Shymkent)	168
story	Story for language learners	404
wikitravel	Phrases from Wikitravel	177
Өлген_қазан	Folk tale from Wikisource	134
wikipedia	Random sentences from Wikipedia	559
Ер_төстік	Folk tale from Wikisource	206
Жиырма_Бесінші_Сөз	Philosophical text	435
		2071

TABLE III: Test data

B. Training the model

We used SRILM toolkit [55, 56] to estimate root and IG probabilities $\Pr(r^l|r^m)$ and $\Pr(g^l|g^m)$ respectively. We need to say few words about the way we prepared root and IG sequences for feeding into SRILM. First of all we used the following tags from the Apertium tagset to split analyzes across the derivational boundaries: `<subst>` (substantive, like a noun), `<attr>` (attributive, like an adjective), `<advl>` (adverbial, like an adverb), `<ger_*>` (gerunds in different tenses), `<gpr_*>` (verbal adjectives in different tenses), `<gna_*>` (verbal adverbs in different tenses), `<prc_*>` (participles in different tenses), `<ger>` (gerund)[5]. Now assume that using the notation from Section 10 the hand-annotated (or unambiguous) text chunk of length n is represented as $\{(r_i, g_{i,1}, \ldots, g_{i,n_i})\}_{i=1}^{n}$. Then we form root-bigrams as

$$(r_1, r_2), (r_2, r_3), \ldots, (r_{i-1}, r_i), \ldots, (r_{n-1}, r_n),$$

and we form IG-bigrams as follows:

$$\begin{aligned}
&(g_{1,n_1}, g_{2,1}), (g_{1,n_1}, g_{2,2}), \ldots, (g_{1,n_1}, g_{2,n_2}),\\
&(g_{2,n_2}, g_{3,1}), (g_{2,n_2}, g_{3,2}), \ldots, (g_{2,n_2}, g_{3,n_3}),\\
&\ldots\\
&(g_{i-1,n_{i-1}}, g_{i,1}), (g_{i-1,n_{i-1}}, g_{i,2}), \ldots, (g_{i-1,n_{i-1}}, g_{i,n_i}),\\
&\ldots
\end{aligned}$$

[4]available at `http://wiki.apertium.org/wiki/Annotation_guidelines_for_Kazakh`

[5]a detailed description of Turkic tagset in Apertium project is given at `http://wiki.apertium.org/wiki/Turkic_lexicon`

The way we form the above bigrams is dictated by the assumptions from Section IV that the root in the current parse depends only on the root of the previous parse, and each inflectional group in the current parse depends only on the last inflectional group of the previous parse.

C. Results

Once our model was trained, i.e. its parameters were estimated, we analyzed the test data with *apertium-kaz* [52] and applied the Algorithm 1 to its output. The accuracy results are given in the column 'Tagger' of the Table IV. As one can see the performance of this purely statistical approach is barely satisfactory (e.g. compared to state of the art for Turkish [38]). This is mainly due to relatively small amount of available hand-tagged corpora for Kazakh. However, if we preprocess the output of the transducer using CG-rules [4] and then just select the first analysis for each ambiguous token, then the accuracy is around 87% on our test set (see column 'CG' in Table IV), which is comparable to the previous results [41, 43] for Kazakh morphological disambiguation. Combining rule-based and statistical approaches, i.e. preprocessing the transducer's output with CG and then selecting most probable parses based on statistical model, yields around 91% accuracy (see column 'CG+Tagger' in Table IV). However, keep in mind that for the fair comparison

Document	Tagger	CG	CG+Tagger
Шымкент	88.46	89.74	92.95
story	76.49	84.16	88.61
wikitravel	71.75	80.23	87.57
Өлген_қазан	88.81	88.06	91.79
wikipedia	93.92	93.56	95.89
Ер_Төстік	85.92	83.01	91.26
Жиырма_Бесінші_Сөз	81.84	85.52	85.98
TOTAL	84.55	87.20	90.73

TABLE IV: Accuracy results in %

of our approach with the previously developed methods one needs to use the same tagset and to test against the same data, which is currently not feasible since both previous works on morphological disambiguation for Kazakh ([41] and [43]) have released neither their tools nor their data for open access.

Let us perform an example of error analysis for the 'CG+Tagger' configuration. One of the most common errors was that it was choosing `<n><nom>` instead of `<n><attr>`: e.g. in

және *көрші* *аймақтардың* 'and of neigboring regions'
`<cnjcoo>` `<n><attr>` `<n><pl><gen>`

the word *көрші* 'neighbor' was mistakenly tagged as `<n><nom>`. A closer look at IG log-probabilities reveals:

$$\log \Pr(\texttt{n}|\texttt{cnjcoo}) = -1.617432$$
$$\log \Pr(\texttt{attr}|\texttt{cnjcoo}) = -1.485425$$
$$\log \Pr(\texttt{n.pl.gen}|\texttt{attr}) = -1.808777$$
$$\log \Pr(\texttt{n.nom}|\texttt{cnjcoo}) = -0.7627025$$
$$\log \Pr(\texttt{n.pl.gen}|\texttt{n.nom}) = -3.236619$$

and we can see that although there are more chances to see a noun in a non-possesive form after an attributive noun than after a noun in nominative case, due to split of the analysis `<n><attr>` into two inflectional groups the wrong parse gets higher overall probability:

$$\begin{aligned}
&\Pr(\texttt{cnjcoo}, \texttt{n.attr}, \texttt{n.pl.gen}) \\
&= \underbrace{\Pr(\texttt{n}|\texttt{cnjcoo}) \Pr(\texttt{attr}|\texttt{cnjcoo}) \Pr(\texttt{n.pl.gen}|\texttt{attr})}_{10^{-4.911634}} \\
&< \underbrace{\Pr(\texttt{n.nom}|\texttt{cnjcoo}) \Pr(\texttt{n.pl.gen}|\texttt{n.nom})}_{10^{-3.9993215}} \\
&= \Pr(\texttt{cnjcoo}, \texttt{n.nom}, \texttt{n.pl.gen})
\end{aligned}$$

This observation leads to a following suggestion: maybe we should try not splitting `<n><attr>` but rather treating it as `<adj>` (an adjective) during the training and tagging. Since we can always distinguish between noun/adjective in Kazakh [57] then theoretically a word cannot have both `<n><attr>` and `<adj>` as possible analyzes, and thus our suggested replacement can be back-substituted without causing any additional ambiguity. This might also work for other errors as well, e.g. when the tagger mistakenly prefers `<adv>` (adverb) over `<adj><advl>` (adverbial adjective) or `<n>` (noun) over `<adj><subst>` (substantivized adjective) and etc.

The list of most common errors for the 'CG+Tagger' configuration also includes

selecting:	instead of:
`<n><nom>` (noun)	`<np><ant><m><nom>` (proper noun)
`<cnjcoo>` (conjucntion)	`<prn><itg><nom>` (inter. pronoun)
`<det><dem>` (dem. determiner)	`<prn><dem><nom>` (dem. pronoun)
`<prn><dem><pl><nom>`	`<prn><pers><p3><pl><nom>`
`<v><tv><aor><p3><pl>`	`<v><tv><aor><p3><sg>`

VI. Conclusion and future work

We reproduced the previous methods of statistical morphological disambiguation [5] for the case of Kazakh language in terms of the Apertium tagset. Combining rule-based and statistical approaches we were able to achieve better accuracy than when these approaches were used separately in the task of morphological disambiguation for Kazakh language. Both the tagger and the annotated data are free and available in open access.

In the future, we are planning to improve the performance of the tagger by adding more annotated data and taking into account suggestions from the previous section. Then our result will directly be able to feed into other work on Kazakh language technology, such as machine translation. Assylbekov and Nurkas [8] made use of the partially-disambiguated output of the morphological analyser to preprocess the Kazakh side of a parallel corpus for statistical machine translation (SMT), achieving an increase in translation quality. We expect that better disambiguation of the analyzer's output will lead to improved performance of the SMT system. We are also planning to apply our disambiguation tool to reduce data sparseness in the task of document and sentence alignment between

Kazakh and English or Kazakh and Russian: given accurate transducers and disambiguation tools for English and Russian, we can apply morphological analysis and then morphological disambiguation to both sides of a candidate pair and then compare the stems in both documents to compute content-based similarity in addition to structural similarity measures as it was done in [58, 59, 60, 61].

Where to find the hand-tagged texts and the tagger

Our morphological disambiguation tool (including hand-annotated texts) is under GNU General Public License (GPL) version 3.0[6]: its code and releases can be found at `https://svn.code.sf.net/p/apertium/svn/branches/kaz-tagger/`.

Aknowledgements

We would like to thank Daiana Azamat for assisting in hand-annotation of the texts and rigorous derivation of the maximum likelihood estimates (13).

References

[1] D. Yuret and E. Biçici, "Modeling morphologically rich languages using split words and unstructured dependencies," in *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, 2009, pp. 345–348.

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

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

[4] I. Salimzyanov, J. Washington, and F. Tyers, "A free/open-source Kazakh-Tatar machine translation system," *Machine Translation Summit XIV*, 2013.

[5] 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.

[6] F. Karlsson, A. Voutilainen, J. Heikkilae, and A. Anttila, *Constraint Grammar: a language-independent system for parsing unrestricted text*. Walter de Gruyter, 1995, vol. 4.

[7] M. P. Lewis, F. Gary, and D. Charles, "Ethnologue: Languages of the world,. dallas, texas: Sil international. retrieved on 15 april, 2014," 2013.

[8] Z. Assylbekov and A. Nurkas, "Initial explorations in kazakh to english statistical machine translation," in *The First Italian Conference on Computational Linguistics CLiC-it 2014*, 2014, p. 12.

[9] N. Habash and F. Sadat, "Arabic preprocessing schemes for statistical machine translation," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, 2006, pp. 49–52.

[10] A. Bisazza and M. Federico, "Morphological pre-processing for turkish to english statistical machine translation." in *IWSLT*, 2009, pp. 129–135.

[11] C. Mermer, "Unsupervised search for the optimal segmentation for statistical machine translation," in *Proceedings of the ACL 2010 Student Research Workshop*. Association for Computational Linguistics, 2010, pp. 31–36.

[12] E. Bekbulatov and A. Kartbayev, "A study of certain morphological structures of kazakh and their impact on the machine translation quality," in *Application of Information and Communication Technologies (AICT), 2014 IEEE 8th International Conference on*. IEEE, 2014, pp. 1–5.

[13] M. L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J. A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. M. Tyers, "Apertium: a free/open-source platform for rule-based machine translation," *Machine translation*, vol. 25, no. 2, pp. 127–144, 2011.

[14] N. Chomsky, *Syntactic structures*. Walter de Gruyter, 2002.

[15] S. Klein and R. F. Simmons, "A computational approach to grammatical coding of english words," *Journal of the ACM (JACM)*, vol. 10, no. 3, pp. 334–347, 1963.

[16] G. Salton and R. Thorpe, "An approach to the segmentation problem in speech analysis and language translation," in *Proceedings of the 1961 International Conference on Machine Translation of Languages and Applied Language Analysis*, vol. 2. Citeseer, 1962, pp. 703–724.

[17] B. B. Greene and G. M. Rubin, *Automatic grammatical tagging of English*. Department of Linguistics, Brown University, 1971.

[18] W. S. Stolz, P. H. Tannenbaum, and F. V. Carstensen, "Stochastic approach to the grammatical coding of english," *Communications of the ACM*, vol. 8, no. 6, pp. 399–405, 1965.

[19] R. Garside, G. Sampson, and G. Leech, *The computational analysis of English: A corpus-based approach*. Longman, 1988, vol. 57.

[20] I. Marshall, "Tag selection using probabilistic methods," *The Computational analysis of English: a corpusbased approach*, pp. 42–65, 1987.

[21] K. W. Church, "A stochastic parts program and noun phrase parser for unrestricted text," in *Proceedings of the second conference on Applied natural language processing*. Association for Computational Linguistics, 1988, pp. 136–143.

[22] S. J. DeRose, "Grammatical category disambiguation by statistical optimization," *Computational Linguistics*, vol. 14, no. 1, pp. 31–39, 1988.

[23] J. Kupiec, "Robust part-of-speech tagging using a hidden markov model," *Computer Speech & Language*, vol. 6,

[6] `http://www.gnu.org/licenses/gpl-3.0.html`

no. 3, pp. 225–242, 1992.

[24] 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.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[26] B. Merialdo, "Tagging english text with a probabilistic model," *Computational linguistics*, vol. 20, no. 2, pp. 155–171, 1994.

[27] E. Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging," *Computational linguistics*, vol. 21, no. 4, pp. 543–565, 1995.

[28] W. Daelemans, J. Zavrel, P. Berck, and S. Gillis, "Mbt: A memory-based part of speech tagger-generator," *arXiv preprint cmp-lg/9607012*, 1996.

[29] A. Ratnaparkhi *et al.*, "A maximum entropy model for part-of-speech tagging," in *Proceedings of the conference on empirical methods in natural language processing*, vol. 1. Philadelphia, USA, 1996, pp. 133–142.

[30] M. Levinger, A. Itai, and U. Ornan, "Learning morpho-lexical probabilities from an untagged corpus with an application to hebrew," *Computational Linguistics*, vol. 21, no. 3, pp. 383–404, 1995.

[31] J. Hajič and B. Hladká, "Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset," in *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 483–490.

[32] J. Hajič, "Morphological tagging: Data vs. dictionaries," in *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*. Association for Computational Linguistics, 2000, pp. 94–101.

[33] N. Ezeiza, I. Alegria, J. M. Arriola, R. Urizar, and I. Aduriz, "Combining stochastic and rule-based methods for disambiguation in agglutinative languages," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 380–384.

[34] B. Megyesi, "Improving brill's pos tagger for an agglutinative language," in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999, pp. 275–284.

[35] K. Oflazer and İ. Kuruöz, "Tagging and morphological disambiguation of turkish text," in *Proceedings of the fourth conference on Applied natural language processing*. Association for Computational Linguistics, 1994, pp. 144–149.

[36] K. Oflazer and G. Tur, "Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation," *arXiv preprint cmp-lg/9604001*, 1996.

[37] K. Oflazer and G. Tür, "Morphological disambiguation by voting constraints," in *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 1997, pp. 222–229.

[38] H. Sak, T. Güngör, and M. Saraçlar, "Morphological disambiguation of turkish text with perceptron algorithm," in *Computational Linguistics and Intelligent Text Processing*. Springer, 2007, pp. 107–118.

[39] D. Yuret and F. Türe, "Learning morphological disambiguation rules for turkish," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 328–334.

[40] M. Kutlu and I. Cicekli, "A hybrid morphological disambiguation system for turkish." in *IJCNLP*, 2013, pp. 1230–1236.

[41] O. Makhambetov, A. Makazhanov, I. Sabyrgaliyev, and Z. Yessenbayev, "Data-driven morphological analysis and disambiguation for kazakh," in *Computational Linguistics and Intelligent Text Processing*. Springer, 2015, pp. 151–163.

[42] O. Makhambetov, A. Makazhanov, Z. Yessenbayev, B. Matkarimov, I. Sabyrgaliyev, and A. Sharafudinov, "Assembling the kazakh language corpus." in *EMNLP*, 2013, pp. 1022–1031.

[43] G. Kessikbayeva and I. Cicekli, "A rule based morphological analyzer and a morphological disambiguator for kazakh language," 2016.

[44] E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz, "Equations for part-of-speech tagging," in *AAAI*, 1993, pp. 784–789.

[45] J. L. Lagrange, *Mécanique analytique*. Mallet-Bachelier, 1853, vol. 1.

[46] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3-4, pp. 237–264, 1953.

[47] W. A. Gale and G. Sampson, "Good-turing frequency estimation without tears*," *Journal of Quantitative Linguistics*, vol. 2, no. 3, pp. 217–237, 1995.

[48] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *Information Theory, IEEE Transactions on*, vol. 37, no. 4, pp. 1085–1094, 1991.

[49] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1. IEEE, 1995, pp. 181–184.

[50] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer*

Speech & Language, vol. 13, no. 4, pp. 359–393, 1999.
[51] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999, vol. 999.
[52] J. N. Washington, I. Salimzyanov, and F. M. Tyers, "Finite-state morphological transducers for three Kypchak languages," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC*, 2014.
[53] S. O. Rojas, M. L. Forcada, and G. R. Sánchez, "Construcción y minimización eficiente de transductores de letras a partir de diccionarios con paradigmas," *Procesamiento del lenguaje natural*, vol. 35, pp. 51–57, 2005.
[54] F. M. Tyers and J. Washington, "Towards a free/open-source universal-dependency treebank for Kazakh," in *3rd International Conference on Computer Processing in Turkic Languages (TURKLANG 2015)*, 2015.
[55] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit." in *INTERSPEECH*, vol. 2002, 2002, p. 2002.
[56] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "Srilm at sixteen: Update and outlook," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop*, 2011, p. 5.
[57] B. KREJCI and L. GLASS, "The kazakh noun/adjective distinction."
[58] Y. Zhang, K. Wu, J. Gao, and P. Vines, "Automatic acquisition of chinese–english parallel corpus from the web," in *Advances in Information Retrieval*. Springer, 2006, pp. 420–431.
[59] M. Esplà-Gomis and M. Forcada, "Combining content-based and url-based heuristics to harvest aligned bitexts from multilingual sites with bitextor," *The Prague Bulletin of Mathematical Linguistics*, vol. 93, pp. 77–86, 2010.
[60] I. San Vicente and I. Manterola, "Paco2: A fully automated tool for gathering parallel corpora from the web." in *LREC*, 2012, pp. 1–6.
[61] L. Liu, Y. Hong, J. Lu, J. Lang, H. Ji, and J. Yao, "An iterative link-based method for parallel web page mining," *Proceedings of EMNLP*, pp. 1216–1224, 2014.