information

MDPI

Article

Research on Uyghur Pattern Matching Based on Syllable Features

Wayit Abliz [1,2], Maihemuti Maimaiti [1,2], Hao Wu [2], Jiamila Wushouer [1,2], Kahaerjiang Abiderexiti [1,2], Tuergen Yibulayin [1,2] and Aishan Wumaier [1,2,*]

[1] School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; wayit@xju.edu.cn (W.A.); mahmutjan@xju.edu.cn (M.M.); jamila@xju.edu.cn (J.W.); kaharjan@xju.edu.cn (K.A.); turgun@xju.edu.cn (T.Y.)

[2] Key Laboratory of Multilingual Information Technology in Xinjiang Uygur Autonomous Region, Urumqi 830046, China; wuhao1994830@163.com

* Correspondence: hasan1479@xju.edu.cn; Tel.: +86-136-599-13514

Received: 27 March 2020; Accepted: 1 May 2020; Published: 2 May 2020

Abstract: Pattern matching is widely used in various fields such as information retrieval, natural language processing (NLP), data mining and network security. In Uyghur (a typical agglutinative, low-resource language with complex morphology, spoken by the ethnic Uyghur group in Xinjiang, China), research on pattern matching is also ongoing. Due to the language characteristics, the pattern matching using characters and words as basic units has insufficient performance. There are two problems for pattern matching: (1) vowel weakening and (2) morphological changes caused by suffixes. In view of the above problems, this paper proposes a Boyer–Moore-U (BM-U) algorithm and a retrievable syllable coding format based on the syllable features of the Uyghur language and the improvement of the Boyer–Moore (BM) algorithm. This algorithm uses syllable features to perform pattern matching, which effectively solves the problem of weakening vowels, and it can better match words with stem shape changes. Finally, in the pattern matching experiments based on character-encoded text and syllable-encoded text for vowel-weakened words, the BM-U algorithm precision, recall, F1-measure and accuracy are improved by 4%, 55%, 33%, 25% and 10%, 52%, 38%, 38% compared to the BM algorithm.

Keywords: pattern matching; text search; Uyghur; syllable; Boyer–Moore; BM-U

1. Introduction

Pattern matching refers to a given string (hereinafter referred to as text) T with length n, and another string (hereinafter referred to as pattern) P with length m ($m \leq n$). It is necessary to find out the starting position of the first occurrence or all occurrences of pattern P in text T. Once found, it is called a success; otherwise, the match fails. Pattern matching is one of the basic research contents of computer science [1]. As an important text processing technology, pattern matching has been applied to many related studies, such as data processing, data compression, text editing, machine translation, search engines, virus and network intrusion detection, content filtering and genetic detection etc. [2–9]. The quality of pattern matching will directly affect the quality of related research and the complexity of the algorithm.

According to China's sixth census in 2010, the Uyghur population is 10 million, and the language belongs to a low-resource language. From a technical point of view, since Windows Vista, iOS 8.0 and Android 4.0 operating systems began to fully support Uyghur language from the system level, the Uyghur network text resources and information to be processed also expanded rapidly, which also accelerated the Uyghur natural language processing (NLP) progress. In April 2019, Tencent

launched a machine translation tool containing Uyghur-Chinese translation based on the WeChat platform. In February 2020, Google Translation also added Uyghur language support. At present, the agglutinative and morphological complexity of Uyghur language is one of the main difficulties in its pattern matching research.

In languages such as English, Chinese, and Uyghur, characters and words are constituent units of different granularities in the language and are often used as the basic unit of pattern matching research. With the large-scale growth of textual information and content in the Uyghur language, higher requirements have been placed on the technical processing of pattern matching in Uyghur. When researching the pattern matching of Uyghur, due to the language characteristics, there are rich morphological changes, and different suffixes can form new words by splicing. Therefore, the research on word pattern matching in Uyghur faces two problems: (1) research using characters as the basic unit, with each word composed of multiple characters, and there is a low matching efficiency; and (2) when the word is used as the basic unit for matching. The morphological complexity of words leads to low matching efficiency.

By analyzing the Uyghur word structure and morphological changes, almost all words can be composed of a certain number of syllables. Therefore, this paper considers designing a data format of syllables as the basic data unit to study the single pattern matching task in Uyghur. The Boyer–Moore algorithm was improved by combining the syllable feature information of the morphological changes of words, and pattern matching was performed on the ordinary text and the syllable-encoded text proposed in this paper. Experimental results show that the method has good performance.

The main contributions of this paper are as follows. (1) Our research on the structural features of Uyghur words and syllables, and the proposed searchable compression format based on syllables, will help improve the performance of existing pattern matching algorithms. (2) We conducted in-depth research on the morphological changes of words caused by weakened vowels. Through the limited expansion of pattern matching sequences, the problem of mismatch caused by morphological changes is solved, and the semantically similar matching effect and recall, precision, accuracy, and F1 values have improved significantly. (3) The research on pattern matching in this paper is also applicable to other syllabic agglutinative languages and can serve as a useful reference for the pattern matching research of other languages of the same type.

2. Related Research

The related algorithms of pattern matching matured in the past few decades, and several classic algorithms have appeared [10,11]. Subsequent improvements have been made to these algorithms [12,13]. At present, the research on pattern matching pays more attention to application innovation and improvement in specific tasks, such as NLP, information retrieval, text filtering and network security. In Uyghur, the study of pattern matching started late.

Syllables, as one of the main Uyghur features, have been widely studied in recent years. Research based on syllable feature information covers tasks such as speech recognition, speech synthesis, lexical analysis, named entity recognition, and spell checking [13–18]. A multi-pattern matching algorithm for Uyghur was researched for the first-time using syllable information [19]. This method uses the number and structure of syllables as one of the matching conditions to improve the matching efficiency. However, this method can only match words with consistent morphological features and cannot match words with weakened vowels and changed syllable structures. Syllable segmentation and analysis of syllable structure are required during the matching process. The Uyghur text filtering task has also been studied [20]; the authors used extended stems and an additional suffixes library to improve pattern matching performance and deal with vowel weakening.

There are also many studies on pattern matching in compressed formats. The corresponding pattern matching algorithms for different compression units and compression algorithms are also different. Usually, the short text [21], the suffix [22,23], the word [24], and the character string [25,26] are used as the pattern matching unit of the compressed content. Some studies have used the BM

algorithm as a pattern matching algorithm in compression format [27,28]. Narupiyakul [29] and Paul G [30] treats syllables as the retrieval unit.

3. Uyghur Language

3.1. Uyghur Alphabet

The current Uyghur language is based on Arabic alphabets, with a total of 32 characters, including 24 consonants "ب/ b", "پ/ p", "ت/ t", "ج/ j", "چ/ č ", "خ/ x ", "د/ d ", "ر/ r ", "ز/ z ", "ژ/ ž ", "س/ s ", "ش/ š ", "غ/ ğ ", "ف/ f ", "ق/ q ", "ك/ k ", "گ/ g ", "ڭ/ ŋ ", "ل/ l ", "م/ m ", "ن/ n ", "ھ/ h", "ۋ/ w", "ي/ y", 8 vowel characters "ا/ a", "ە/ ä", "ې/ e", "ى/ i", "و/ o ", "ۆ/ ö ", "ۇ/ u ", "ۈ/ ü ", and a special symbol Hamza (u0626). The encoding range is in the Unicode basic area (U0600–U06FF), occupying 2 bytes, and the writing direction is from right to left. This paper uses Latinized transliteration to represent Uyghur letters.

3.2. Morphological Changes of Words

Uyghur is a typical agglutinative language. It has strong derivational ability and rich morphological variations. The complex morphology of words is the main feature of the agglutinative language [31–35]. As a typical complex agglutinative language, its morphological structure is word = stem + [suffix]. There are two types of suffixes: the inflectional suffix and the derivational suffix. Adding roots or stems the derivational suffix generates new words, similar to *work* + *man* = *workman*. After the inflectional suffix is added, it only changes the grammatical attributes such as the person, plural, and case of the original word. Similar to *book* + *s* = *books*, this paper discusses the inflectional suffix. Uyghur noun stems can be connected with different suffixes and support continuous concatenation of multiple suffixes. For example, the noun "قويۇڭلارنىڭ" "*qoyuŋlarniŋ*" is generated by adding three layers of suffixes to the stem *qoy* (sheep): (1) *qoy* + *uŋ* (your sheep); (2) *qoyuŋ* + *lar* (your sheep, sheep plural); (3) *qoyuŋlar* + *niŋ* (your sheep's, sheep plural);

3.3. Vowel Weakening

Modern Uyghur phonetic harmony is very common, and one of the main manifestations is the weakening of vowels. Vowel weakening refers to the weakening of vowels into other vowels when some additional elements are added to the stem composed of specific vowels, such as *Är (man) + i (third person) = Eri (his man Ä→E); karwat (bed) + im (first person) = karwitim (my bed, a→i); Taš (stone) + iŋ (second person) = tešiŋ (your stone, a→e).*

Mireguli et al. [36] proposed an algorithm to identify the Uyghur vowel weakening based on the word and syllable structure. Other languages have similar situations [37–44]. Uyghur vowel weakening occurs frequently in written form. There are special exceptions, such as *Taj (crown) + i (third person) = Taji (crown, a→a).* The weakening rules are complex, and all phenomena cannot be described completely according to the rules. In the 27,266 stem words collected from the orthographic dictionary, 13,843 (50.7%) are structurally weak vowels [45]. Although these words contain a certain amount of irregular words, it can be seen that the weakening of vowels is a very common phenomenon in Uyghur.

3.4. Syllable-Encoded Text

There are no special signs between Uyghur syllables. The pronunciation of syllables alone and in words is unchanged [14]. There are 12 types of syllables in current Uyghur words, with C for consonant and V for vowel. The syllable types are the six syllable structures V, VC, CV, CVC, VCC, and CVCC. Meanwhile, CCV, CCVC, CCVCC, CVV, CVVC, and CCCV are structures for recording foreign words. The CVV and CVVC structures with two Vs are used for Chinese or other language words with two vowels. This paper uses the syllable segmentation method described by Wayit et al. [46]. Wayit et al. [47] found that the top 2000 Uyghur syllables with the highest frequency can cover 99% of words, and proposed a syllable coding scheme B16 encodes each syllable, in which a syllable is encoded in the

same length as the Unicode character encoding length. The encoding area is within the Unicode Private Use Area (ue000–uf8ff). This paper uses the Wayit et al. [47] coding scheme to design a text format based on syllable encoding and changes the basic unit of string pattern matching in text from the original characters to syllables to compress strings while achieving syllable-based pattern matching.

4. Uyghur String Matching

4.1. Basic Concepts

There are several search-related symbols and supplementary definitions for string matching used in this paper:

1. *uChar* is a Uyghur Unicode character, and its encoding range is (u0600–u06ff).
2. *Sb* is a syllable, which is composed of several *uChar*. When *Sb* is a syllable composed of three *uChar*, its structure is *Sb [uChar1 uChar2 uChar3]*.
3. *Sc* is the syllable encoding of *Sb* in B16 encoding scheme [47]. Each *Sc* encoding length is equal to a Unicode character, and the encoding range is in the Unicode Private Use Area (ue000–uf8ff).
4. *W* is a Uyghur word composed of several *Sb*, *W (Sb1Sb2 . . . Sbn)*, its length is equal to the number of *uChar* in the word.
5. *Wz* is the result of syllable segmentation and encoding of word *W*. When *W* has three syllables, its structure is *Wz (Sc1Sc2Sc3)*, and the length of *Wz* is equal to the number of syllables of *W*.
6. *P* is a pattern and noun stem, its structure is similar to *W*, and *W* = *P* + *Inflectional suffix*.
7. *Pz* is the syllable code of pattern *P*, and its structure is similar to *Wz*.
8. *T* is a text containing *n W*, its structure is *T (W0, W1, . . . , W (n−1))*.
9. *Tz* is a syllable-encoded compressed text, which is generated by *T* after syllable encoding. Its structure is *Tz (Wz0, Wz1, . . . , Wz(n−1))*.
10. Structure matching: when the sequence of characters in string *S1* is unchanged, it is completely contained in string *S2*, and the length of *S1* ≤ the length of *S2*
11. Semantic matching: when *W* = *P* + *Inflectional suffixes*, semantics of pattern *P* are included in word *W*; then, *P* and *W* have semantic matching. Sometimes, the weakening of vowels results in the change of *W* structure and the mismatch of *P* structure. Pattern *P* length is less than or equal to word *W* length.
12. Matching result: when *P* matches a *W* semantic or structure in *T*, a complete *W* is returned. For example, when *P* = *man*, *T* = *{"other", "manchu", "mankind", "man", "men"}*, the result of *P* structure matching is *{"manchu", "mankind", "man"}*, and the result of semantic matching is *{"man", "mankind", "men"}*.

4.2. Retrieval Parameters and Calculation Formulas

The ideal search results for this article are listed below:

Input: 1 *P* / *Pz*, Output: Returns all *W* / *Wz* related to *P* / *Pz* semantics in *T*/*Tz*

Formulas 1–4 calculate the matching results of recall, precision, accuracy, and F1-measure value. The semantics of the parameters in the formula in this article are as follows.

TP (True Positive): There is a semantic matching relationship between *P*/*Pz* and *W*/*Wz*, and the matching result includes *W*/*Wz*.

TN (True Negative): There is no semantic matching relationship between *P*/*Pz* and *W*/*Wz*, and the matching result does not include *W*/*Wz*.

FP (False Positive): There is no semantic matching relationship between *P*/*Pz* and *W*/*Wz*. The matching result includes *W*/*Wz*.

FN (False Negative): There is a semantic relationship between *P*/*Pz* and *W*/*Wz*, and the matching result does not include *W*/*Wz*.

The higher the TN and TP, the better, and the lower the FP and FN, the better. The total number of samples is TP + TN + FP + FN.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$\text{F1} = \frac{2 \times P \times R}{P + R}. \tag{4}$$

4.3. Preparation of Experimental Corpus

Because of the morphological complexity of agglutinative language words, the usual pattern matching experiment method prepares a certain size of corpus T and then randomly selects P of different lengths or selects a certain number of highly related words as P (e.g., the top 10 words with the highest correlation degrees). This method does not ensure that all forms of a word can be matched because some words change their structure more than once after adding a suffix, such as *naxša+lar→naxšilar+iŋ→naxšiliriŋ (ša→ši, lar→lir)*. Moreover, some forms of a word rarely appear, and the experimental corpus T may not include this form. This article prepares three types of experimental corpus.

1. Type A corpus generated by the algorithm. First, this paper selects 22 high-frequency words based on word length, syllable structure, and number of syllables (Table 1); 11 of the words are weakened (Word V.W). These weakened words cover all four vowel weakening types (a→i, a→e, ä→i, ä→e). We design a morphology-based word generator algorithm based on stemming, taking nouns as an example, this algorithm can generate all the 312 forms of P in a dictionary [45] by adding 1–4 layers of suffixes. The experimental corpus T generated by this algorithm covers 22 × 312 = 6864 forms of 22 words. If the matching algorithm can match all 312 forms of P according to the P, it means that the algorithm theoretically can recognize and match all forms of pattern P in any natural language environment, and hence recall = 1. When recall = 1, corpora B and C can be used for the next experiment to test the pattern matching ability of the algorithm in a natural language environment.
2. Type B corpus made from natural language text. We collect a certain amount of actual corpus for word segmentation to generate a word data set. The content of the dataset is the words appearing in the corpus and the frequency of occurrence F in the corpus. The corpus is based on Unicode-encoded text with a size of 46.8 MB and has covered comprehensive news, agricultural technology, agency names, novels, natural sciences, dictionaries and encyclopedias, and social media short texts. There were 136,523 unique words and 7434 unique syllables.
3. Type A and type B corpus is a list of experimental words obtained through algorithm derivation and database fuzzy query; type C is a paragraph of natural language text composed of several typical sentences.

Table 1. Experimental words and syllable structure attributes in type A corpus.

No	Word		Syllable	Word (V.W)		Syllable	V.W. e.g.,
1	dog	it	vc	Man	**ä**r	vc	eri[ä→e]
2	bird	quš	cvc	Tea	č**a**y	cvc	čeyi[a→e]
3	year	yil	cvc	Land	y**ä**r	cvc	yeri[ä→e]
4	sue	ärz	vcc	Ideal	ğay**ä**	cv+cv	ğay**i**ni[ä→i]
5	chiken	toxu	cv+cv	Ox	kal**a**	cv+cv	kal**i**si[a→i]
6	people	xälq	cvcc	Human	ad**ä**m	v+cvc	ad**i**mi[ä→i]
7	woman	ayal	v+cvc	Apple	alm**a**	vc+cv	alm**i**lar[a→i]

Table 1. *Cont.*

No	Word		Syllable	Word (V.W)		Syllable	V.W. e.g.,
8	river	därya	cvc+cv	Paper	qäğ**ä**z	cv+cvc	qäğ**i**zi[ä→i]
9	Roosevelt	Rozwelt	cvc+cvcc	Ethnic	mill**ä**t	cvc+cvc	mill**i**ti[ä→i]
10	product	mähsulat	cvc+cv+cvc	America	Amerik**a**	v+cv+cv+cv	amerik**i**da[a→i]
11	airplane	ayrupilan	vc+cv+cv+cvc	Song	naxš**a**	cvc+cv	naxš**i**niŋ [a→i]

Table 2 shows the statistical information of pattern *P* in the corpus type B. In the table, *P* indicates a test stem, *Pm* indicates the number of words that fuzzy match the *P* in structure, *Pv* is the weakened form of pattern *P*, *Pvm* is the number of words that fuzzy match the *Pv* in structure, *F0* is the occurrence frequency of all *Pm* and *Pvm* in the corpus, and *Pr* is the number of words related to pattern *P* in semantics. For example, when *P* = *är* (man),*ärkäk* (male) belongs to *Pr*, *ärkin* (freedom) does not belong to *Pr*, the labeling of *Pr* is done manually, *Pr* is a part of *Pm* and *Pvm*. *Fr* is the frequency of all *Pr* in the corpus, and *Fr* is a part of *F0*. *Pm* and *Pvm* are obtained through fuzzy query through SQL statements: *select words, frequency from table where words like '%P%'*.

Table 2. Statistics of experimental words in the corpus type B.

No.	P	Pm	Pr	Fr	F0	P	Pm	Pv	Pvm	Pr	Fr	F0
1	it	380	27	302	6077	är	178	eri	264	63	34056	39971
2	quš	236	81	1811	2816	čay	85	čeyi	9	34	421	784
3	yil	672	130	21843	25951	yär	190	yeri	106	209	14876	16788
4	ärz	39	13	144	468	ğayä	4	ğayi	25	25	198	923
5	toxu	46	35	503	527	kala	22	kali	71	37	1934	2562
6	xälq	82	82	11294	11294	adäm	61	adimi	24	85	6091	6091
7	ayal	39	38	17303	17305	alma	118	almi	55	21	87	1796
8	därya	42	42	3776	3776	qäğäz	13	qäğizi	5	18	340	340
9	rozwelt	19	19	1618	1618	millät	22	milliti	10	29	1191	1191
10	mähsulat	60	60	2698	2698	amerika	2	ameriki	34	36	5147	5147
11	ayrupilan	29	29	975	975	naxša	1	naxši	26	27	335	335

4.4. Matching of Existing Algorithms

The Boyer–Moore (BM) algorithm is used to perform pattern matching on experimental corpus type A, and *Tz* is the syllable-encoded text of corpus *T*. Table 3 shows the matching results. In the table, M indicates a successful match, Mis indicates a failed match, and e.g., indicates an example of a failed match. There are three cases of matching status.

1. Both *P* and *T*, *Pz* and *Tz* match exactly, for example: *toxu* and *därya*.
2. *P* and *T* match exactly, and *Pz* and *Tz* partially match, for example: *quš*.
3. Both *P* and *T*, *Pz* and *Tz* have matching failures, for example, *naxša*.

Table 3. Word morphology matching results.

Word (stem)	T (M/Mis)	Tz (M/Mis)	e.g.,	Word (V.W)	T (M/Mis)	Tz (M/Mis)	e.g.,
it	312/0	172/140	i+ti	är	172/140	172/140	***e+ri***
quš	**312/0**	**172/140**	**qu+šum**	čay	172/140	172/140	če+yi
yil	312/0	172/140	yi+li	yär	172/140	172/140	ye+ri
ärz	312/0	172/140	är+zi	ğayä	62/250	1/311	ğayi+si
toxu	**312/0**	**312/0**		kala	62/250	1/311	kali+lar
xälq	312/0	172/140	xäl+qi	adäm	172/140	172/140	adi+mi
ayal	312/0	172/140	aya+li	alma	62/250	1/311	almi+da

Table 3. *Cont.*

Word (stem)	T (M/Mis)	Tz (M/Mis)	e.g.,	Word (V.W)	T (M/Mis)	Tz (M/Mis)	e.g.,
därya	**312/0**	**312/0**		qäğäz	172/140	172/140	qäği+zi
rozwelt	312/0	172/140	rozwelti+niŋ	millät	172/140	172/140	milli+ti
mähsulat	312/0	172/140	mähsula+ti	amerika	62/250	1/311	ameriki+čä
ayrupilan	312/0	172/140	ayrupila+ni	**naxša**	**62/250**	**1/311**	**naxši+si**

5. Improvement of Matching Algorithm

5.1. Analysis

According to experiments, to improve the degree of structural matching between *P* and *T* and between *Pz* and *Tz*, we must first solve the matching failure caused by changes in syllable structure. Below we use '*' for any string, '#' for any string that forms a syllable structure, and sx for any syllable.

1. Changes in syllable structure caused by weakened vowels

The *naxša* in Table 3 is taken as an example. When the third-person suffix *si* is added, the weakening of the vowels results in a change in the morphological structure: *W* = *naxša* + *si* = *naxši* + *si* *(a→i)*. When *P* = *naxša*, the match with *W* = *naxšisi* (his song) fails; in order to be able to retrieve these forms, an algorithm needs to be designed to determine whether vowel weakening may occur based on the morphological structure of *P*, and if so, calculate the pattern *P* weakened form *Pv* and find out the mismatch pattern of P through *Pv*

2. Changes in the syllable structure caused by the addition of suffixes

Taking *Pz* = *quš* in Table 3 as an example, when the first-person suffix um is added, the syllable structure changes as follows: *Wz* = *quš* + *um* → *qu* + *šum* *(cvc* + *vc* →*cv* + *cvc)* (bird→my bird). The syllable structure of *Wz* cannot match *Pz*. If the structural change of *quš* is represented by *quš**, *qu+š#+sx*, then during the matching process, if the algorithm can recognize that the second syllable is a syllable that satisfies š#, the matching problem can be solved. From the syllable structure, š# belongs to C#. According to the Uyghur syllables type, there are five types of structures that may appear: CV, CVC, CVCC, CVV, and CVVC. When the first C is the character š and considering that there are 24 consonants and 8 vowels in Uyghur, then the theoretical type of syllables in the second syllable š# may be:

$$N = šV(1 \times 8) + šVC(1 \times 8 \times 24) + šVCC(1 \times 8 \times 24 \times 24) + šVV(1 \times 8 \times 8) + šVVC(1 \times 8 \times 8 \times 24)$$
$$N = 8 + 192 + 4608 + 64 + 1536 = 6408$$

5.2. Solutions

It is found that the change of the syllable structure occurs between the last syllable of *P* and the first inflectional suffix. According to the rules [45] for attaching suffixes to nouns, *P* adds first layer of suffixes to generate 18 kinds of word forms. For comparison and convenience, we selected *alma* (apple) and *quš* (bird) and added first layer of suffixes to observe the change of the morphological structure. Table 4 shows the additional information.

1. *alma*: there are four structures that can express all other structures within *T* and *Tz*. They are *alma* *\|alma + sx, almam* *\|almam + sx (first person), almaŋ* *\|almaŋ + sx (second person), and almi* *\|almi + sx (third person, weakened).*
2. *quš*: matches all forms of *quš* * in *T* and meets the requirements in *Tz*: *quš +sx* and *qu +š#+sx*. Here, we need to determine the value range of š#. According to the last calculation, š# has 6408 possibilities. By observing š#, there are the following structures: *quš* + *sx* (stem, no person),

qu+ šum+sx (first person), *qu+šung+ sx* (second person), and *qu+ši+sx* (third person). All three forms of *quš* can be represented with three structures, which is a very interesting phenomenon. This means that the value range of *š#* can be reduced from 6408 to only 3 *(šum, šuŋ, ši)* and the remaining 6405 can be ignored. Another exciting result is that if the current value of *š#* is not one of these three, it can be determined that the word *Wz* in *Tz* may not meet the semantic matching condition *Wz = Pz + inflectional suffix*. For example, when *Tz = {Wz1 = so+qu+šuš+niŋ* (the war's …), *Wz2 = tö+gi+qu+ši+niŋ* (the ostrich's …)} because the third syllable of *Wz1 šuš* is not in *(šum, šuŋ, ši)*, this method automatically excludes *Wz1* and can match *Wz2*. When *Pz* is used to search *Tz*, the search results can be used to exclude some words that are not related to *Pz* semantically, without performing a semantic analysis; this further improves the precision, and the retrieval speed is faster. This method is also effective for generating weakened words *alma*. If the structure after *alma* adds a configuration suffix cannot satisfy *(al+mam, al+maŋ, al+mi*), then the matching results are not related to *alma*; for example, *almas (al+mas: diamond)* is not related to *alma* (apple) in semantics.

Table 4. Morphological changes of stem with suffixes.

P+Suffix		**Matching Expression**	
Structure	Results	P	Pv
+Null	alma quš	alma* quš *	al + ma + sx quš + sx
+plural	alm**a** + lar = alm**i**lar quš + lar = qušlar	almi* quš*	al + mi + sx quš + sx
+ Personal	Almam | alm**a** + miz = alm**i**miz | almaŋ | alm**a** + ŋiz = alm**i**ŋiz | alm**a** + si = alm**i**si | alm**a** + liri = alm**i**liri; Qušum | qušumiz | qušuŋ | qušuŋiz | quši | qušliri	almam* almaŋ* almi* quš*	al + mam + sx al + maŋ + sx al + mi + sx quš + sx qu + šum + sx qu + šuŋ + sx qu + ši + sx
+ Case	Alminiŋ | almiğa | almini | almida | almidin | almidäk | almidiki | almiğičä | almičä | almičilik; qušniŋ | qušqa | qušni | qušta | quštin | quštäk | quštiki | qušqičä | quščä | quščilik	almi* quš*	al + mi + sx quš + sx

It can be seen in Table 4 that in order to make recall = 1, two algorithms need to be designed. The first algorithm determines whether *P* satisfies the weakening condition. If it is satisfied, the weakened form *Pv* of *P* is calculated. The second algorithm adds personal suffixes according to the structural characteristics of *P*. The two algorithms finally generate a list *P* for pattern matching, P_{List} = *{P, P1, P2, P3 / Pv}*. Among them, *P1, P2,* and *P3* are the result of adding personal (1–3) suffixes to *P*. The role of P_{List} is to assist the BM algorithm to improve matching efficiency. Because the weakening of vowels is more complicated and cannot be completely solved by rules, there are some special cases not subject to rules or phenomena: for example, tağ + I → teği (a → e, subject to rules), taš + I → teši (a → e, subject to rules), and taj + i → taji (a → a, not subject to rules). The word weakening algorithm for these special cases is solved by adding a special case library.

5.3. Improvement of BM Algorithm

According to the above analysis, if we use the weakening processing algorithm and the suffix addition algorithm to calculate the matching pattern list P_{List} according to *P*, then we can calculate the common part P_{common_part} of the P_{List} as the matching pattern of the BM algorithm. When the algorithm matches one P_{common_part}, it uses the remaining P_{remain_parts} to match. If the match is successful, it starts to find the next P_{common_part}. For a single syllable *P* with weakening, P_{common_part} = *null* may appear. At this time, the algorithm will match each pattern *P* in the P_{List} independently. For example, when *P* =

At (horse), *Pv* = *Eti* (A→E), there is a common symbol Hamza in *T*, and there is no common syllable in *Tz*. The improved new algorithm BM-U is shown in Algorithm 1:

Algorithm 1. BM-U (P, T) pattern matching algorithm

input: P input a stem
output: match_num

```
match_num←0
start position ←0
if IsVowelWeaken(P) then
    Pv← P shortest vowel weakened form
end if
P_list←append each personal suffix to stem P, Pv
P_list_common_part←get P_list common part
for i = 0 . . . P_list. items. count-1 do
    P_list_remain_parts[i]←P_list[i]-P_list_common_part
end for
if P_list_common_part. length=0 then // No common part
  for i = 0 . . . P_list[i]. length-1 do
    Boyer_Moore (T, start position, P_list[i])
  end for
  return match_num
Else
 do
    found = Boyer_Moore (T, start position, P_list_common_part)
    if found then
       for i = 0 . . . P_list_remain_parts. Length-1 do
            P = P_list_common_part + P_list_remain_parts [i]
            if pattern_match (P) then
                match_num++
                start position ←next start position
                break //find 1, looking for next P_list_common_part
            end if
       end for
       // P remaining parts match failed, looking for next P_list_common_part
       start position ←next start position
    else
       return match_num
    end if
  while   start position < T. length - P_list_common_part. Length
end if
return match_num
```

6. Experiment and Analysis

A total of four experiments were performed.

1. We used the BM-U algorithm to test the word morphology matching ability of pattern *P*, and used type A corpus generated by the algorithm. If recall = 1, it means that the new algorithm can recognize all word forms of pattern P, can use the type B and type C corpus to test the algorithm precision, accuracy, F1-measure, and observe the recall value of the algorithm in the natural corpus environment.
2. We used the BM and BM-U algorithms to test the matching ability of patterns *P* and *Pz* on natural language type B corpora *T* and *Tz*. Observe the matching performance of the two algorithms on

the two encoding formats through experiments and calculate the impact of the algorithm and matching unit changes on the matching rate.

3. In order to facilitate the observation of the matching performance of the new algorithm, two algorithms were used to conduct demonstrative pattern matching experiments on natural language paragraphs using type C corpus.
4. We conducted pattern matching of syllable encoding file *Tz* format for monosyllable and non-syllable strings, compare character-based text *T* with syllable-based text *Tz*.

6.1. BM-U Word Morphology Matching Ability

The experimental method and pattern *P* are the same as those in Table 3. The algorithm is changed to the BM-U algorithm. The experimental result is BM-U algorithm uses pattern *P* and pattern *Pz* to correctly match all 312 forms of *P* and *Pz* in type A corpora *T* and *Tz*, and recall = 1. The new algorithm satisfies the conditions and can be used for pattern matching experiments based on natural language corpus B and C.

6.2. Matching Experiments on Natural Language Words

6.2.1. Experimental Results

The experimental results of BM and BM-U algorithms on type B natural language corpus *T* and *Tz* are shown in Tables 5 and 6. In the table, *Pm* and *Pvm* indicate the number of words that can be fuzzy matched with *P* and *Pv* within *T*. *Pr* is the number of words that are semantically related to *P* (semantic match) in (*Pm* + *Pvm*) and are manually labeled. *Alg* is the type of algorithm. T_P / Tz_P, T_R / Tz_R, T_F / Tz_F, and T_A / Tz_A represent the precision, recall, accuracy, and F1-measure values of the algorithm in *T* and *Tz*.

Table 5. Matching results of words without weakening.

No	P	Pm	Pvm	Pr	Alg	T_P	Tz_P	T_R	Tz_R	T_F	Tz_F	T_A	Tz_A
1	it	380	0	27	BM	0.07	0.10	1.00	0.59	0.13	0.17	0.07	0.59
					BM-U	0.07	0.09	1.00	1.00	0.13	0.16	0.07	0.24
2	quš	236	0	81	BM	0.34	0.46	1.00	0.81	0.51	0.59	0.34	0.61
					BM-U	0.34	0.44	1.00	1.00	0.51	0.61	0.34	0.56
3	yil	672	0	130	BM	0.19	0.43	1.00	0.75	0.32	0.54	0.19	0.76
					BM-U	0.19	0.34	1.00	0.96	0.32	0.50	0.19	0.62
4	ärz	39	0	13	BM	0.33	1.00	1.00	0.46	0.50	0.63	0.33	0.82
					BM-U	0.33	0.46	1.00	0.85	0.50	0.59	0.33	0.62
5	toxu	46	0	35	BM	0.76	0.92	1.00	0.97	0.86	0.94	0.76	0.91
					BM-U	0.76	0.92	1.00	0.97	0.86	0.94	0.76	0.91
6	xälq	82	0	82	BM	1.00	1.00	1.00	0.60	1.00	0.75	1.00	0.60
					BM-U	1.00	1.00	1.00	0.91	1.00	0.96	1.00	0.91
7	ayal	39	0	38	BM	0.97	1.00	1.00	0.74	0.99	0.85	0.97	0.74
					BM-U	0.97	1.00	1.00	1.00	0.99	1.00	0.97	1.00
8	därya	42	0	42	BM	1.00	1.00	1.00	0.93	1.00	0.96	1.00	0.93
					BM-U	1.00	1.00	1.00	0.93	1.00	0.96	1.00	0.93
9	rozwelt	19	0	19	BM	1.00	1.00	1.00	0.95	1.00	0.97	1.00	0.95
					BM-U	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
10	mähsulat	60	0	60	BM	1.00	1.00	1.00	0.63	1.00	0.78	1.00	0.63
					BM-U	1.00	1.00	1.00	0.97	1.00	0.98	1.00	0.97
11	ayrupilan	29	0	29	BM	1.00	1.00	1.00	0.69	1.00	0.82	1.00	0.75
					BM-U	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 6. Matching results of words with vowel weakening.

No	P	Pm	Pvm	Pr	Alg	T_P	Tz_P	T_R	Tz_R	T_F	Tz_F	T_A	Tz_A
1	är	178	264	63	BM	0.31	0.37	0.89	0.87	0.46	0.52	0.71	0.77
					BM-U	0.14	0.23	1.00	0.98	0.25	0.37	0.14	0.52
2	čay	85	9	34	BM	0.32	0.38	0.79	0.79	0.45	0.51	0.31	0.46
					BM-U	0.36	0.43	1.00	1.00	0.53	0.60	0.36	0.52
3	yär	190	106	209	BM	0.91	0.93	0.82	0.79	0.86	0.85	0.81	0.81
					BM-U	0.71	0.80	1.00	0.97	0.83	0.87	0.71	0.80
4	ğayä	4	25	25	BM	0.50	1.00	0.08	0.04	0.14	0.08	0.14	0.17
					BM-U	0.86	1.00	1.00	1.00	0.93	1.00	0.86	1.00
5	kala	22	71	37	BM	0.14	0.30	0.08	0.08	0.10	0.13	0.43	0.56
					BM-U	0.40	0.51	1.00	0.95	0.57	0.66	0.40	0.61
6	adäm	61	24	85	BM	1.00	1.00	0.72	0.66	0.84	0.79	0.72	0.67
					BM-U	1.00	1.00	1.00	0.88	1.00	0.94	1.00	0.88
7	alma	118	55	21	BM	0.02	0.20	0.10	0.05	0.03	0.08	0.22	0.86
					BM-U	0.12	0.34	1.00	1.00	0.22	0.51	0.12	0.77
8	qäğäz	13	5	18	BM	1.00	1.00	0.72	0.72	0.84	0.84	0.72	0.72
					BM-U	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
9	millät	22	10	29	BM	1.00	1.00	0.69	0.63	0.81	0.77	0.69	0.63
					BM-U	1.00	1.00	1.00	0.94	1.00	0.97	1.00	0.94
10	Amerika	2	34	36	BM	1.00	1.00	0.06	0.06	0.11	0.11	0.06	0.15
					BM-U	1.00	1.00	1.00	0.97	1.00	0.99	1.00	0.97
11	naxša	1	26	27	BM	1.00	1.00	0.04	0.04	0.07	0.07	0.04	0.04
					BM-U	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

6.2.2. Analysis of Experimental Results

The pattern matching capabilities of *T* and *Tz* (Table 5) were compared using two algorithms, and the comparison results are shown in Table 6. In the table, VW indicates the calculation result of the vowel-weakened words (Table 6), and Not VW indicates the calculation result of the vowel words that are not weakened (Table 5), No. is the calculation formula number, P, R, A, F indicates precision, recall, accuracy, and F1-measure values. Taking formula No. 1 as an example, $P_{T\_BM\text{-}U}$ indicates that *T* is retrieved using the BM-U algorithm, P_{T_BM} is used to retrieve *T* using the BM algorithm, and ΔP is the BM-U algorithm P-value of each word in Table 5 minus the BM algorithm P-value sum of the increments after. ΔP > 0 indicates that the P-value of the BM-U algorithm is higher than that of the BM algorithm, and ΔP < 0 indicates that the P-value of the BM-U algorithm is lower than that of the BM algorithm. Δ represents the sum of the increments of all n words (here n = 11). In formula No.1 Δ= ΔP, and Avg (Δ) represents the average of the increments. Table 7 shows a comparison of the retrieval capabilities of *T* with two algorithms. The basic matching unit of *T* is a character, and the basic matching unit of *Tz* is a syllable

Table 7. Comparison of Boyer–Moore (BM) and Boyer–Moore-U (BM-U) retrieval of T.

No.	Formula	Not VW		VW	
		Δ	Avg (Δ)	Δ	Avg (Δ)
1	$\Delta P = \sum (P_{T\_BM\_U} - P_{T\_BM})$	0	0	0.39	0.04
2	$\Delta R = \sum (R_{T\_BM\_U} - R_{T\_BM})$	0	0	6.01	0.55
3	$\Delta F = \sum (F_{T\_BM\_U} - F_{T\_BM})$	0	0	3.62	0.33
4	$\Delta A = \sum (A_{T\_BM\_U} - A_{T\_BM})$	0	0	2.74	0.25

The improvement of the algorithm without weakening the words has no effect on the matching efficiency of *T*. The values of T_P, T_F, T_A, and T_R are unchanged. Since the content of *T* is collected by the P fuzzy search *(% P%)* method, T_R = 1. After improving the algorithm, the retrieval efficiency of weakened words significantly improved. The improvement of R, F, and A is very obvious, especially the average increase of R by 55%, mainly because the new algorithm can retrieve the weakened words.

For example, when *P* = *alma*, *T* = *{alma, almisi (his apple)}*, the BM match result = *{alma}*, and the BM-U match result = *{alma, almisi}*. The new algorithm effectively increases the F and A values of T by 33% and 25%, respectively. Compared with R, F, and A values, the increase in P-value is not high (the average increase is 4%).

Table 8 presents a comparison of the retrieval effect of *T* with the current BM algorithm and the retrieval of *Tz* with the BM-U algorithm proposed in this paper. For weakened words, all parameters were significantly increased; meanwhile, for non-weakened words, P, F, and A-value increased, R-value decreased and in the BM algorithm always T_R = 1. The R-value of BM-U on *Tz* is obviously improved once the algorithm is improved, but Tz_R <1 for some words. Here, the decline in R is mainly because the partially misspelled word *T* can still meet the matching conditions, and *Tz* cannot meet the syllable-based matching conditions, resulting in Tz_R <1 for BM-U. Taking No. 3 in Table 5 as an example, *Tz_R = 0.75 (TP = 98, FN = 32)* of the BM algorithm when *P = yil*, Tz_R = *0.96 (TP = 125, FN = 5)* for the BM-U algorithm, and improving the algorithm increases the R-value by 21%. However, there are still five words that change the syllable structure due to misspelling {*(FN = 5)*: (*bir+yildn, yild+din, yill+din, yill+rdin, yi+le+si+ri*)} has not been retrieved. The correct spelling of these five words should be { *bir+yil+din, yil+din, yil+din, yil+lir+din, yi+li+siri* }. The retrieval efficiency of the weakened words is relatively obvious in the algorithm. Figure 1 shows a comparison of the R-values of the weakened words by the two algorithms, and Figure 2 shows a comparison of the F and A-values of the weakened words by the two algorithms.

Table 8. Comparison of Tz retrieval by BM-U algorithm and T retrieval by BM algorithm.

No.	Formula	Not VW		VW	
		Δ	Avg (Δ)	Δ	Avg (Δ)
5	$\Delta P = \sum (P_{Tz\_BM\_U} - P_{T\_BM})$	0.59	0.05	1.11	0.10
6	$\Delta R = \sum (R_{Tz\_BM\_U} - R_{T\_BM})$	−0.41	−0.04	5.7	0.52
7	$\Delta F = \sum (F_{Tz\_BM\_U} - F_{T\_BM})$	0.39	0.04	4.2	0.38
8	$\Delta A = \sum (A_{Tz\_BM\_U} - A_{T\_BM})$	1.1	0.10	4.16	0.38

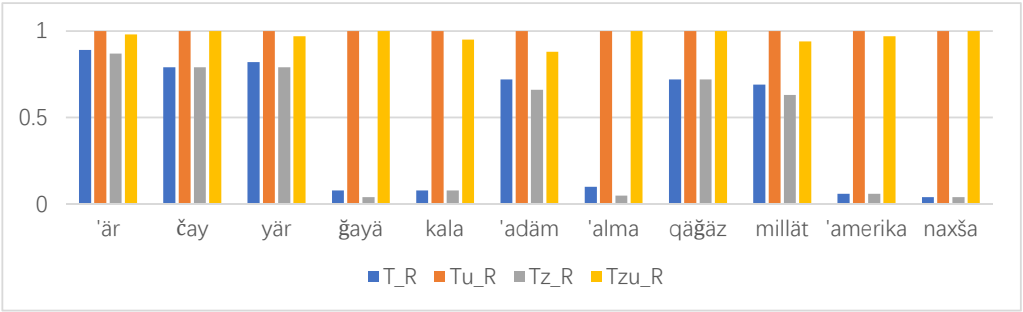

Figure 1. Comparison of BM-U and BM algorithm R-values of weakened words (u means BM-U algorithm).

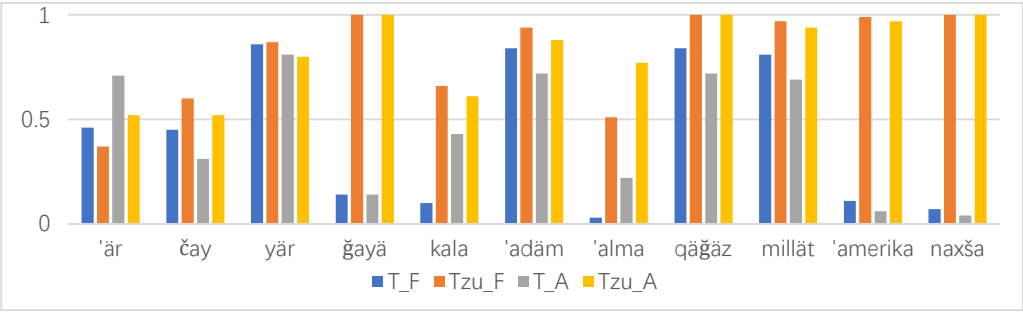

Figure 2. BM-U and BM algorithm F1-measure value (F) and accuracy (A) comparison of weakened words (u means BM-U algorithm).

6.3. Matching Experiments on Natural Language Sentences

Table 9 is the pattern matching results of the two algorithms in character-based natural language sentences, *P = {alma, amerika}.* The improvement of the algorithm allows the BM-U algorithm to match words with weakened vowels and improve user search experience.

Table 9. Pattern matching results based on natural language sentences.

Sample Sentences (Uy/En)	**Keywords (Apple, America)**		**Match**	
Bu almiliq bağdiki almilar bäk oxšaptu. Almaŋ bäk tatliqkän. Almida witamin köp. Alma **Amerika** alma šerkitiniŋ bälgisi. Akam **Amerikidin** maŋa **Amerikida** yasalğan alma telfuni äwätiptu. **Amerikiliq** dostum **Amerikidiki** yuqum sani 500 miŋdin ašti, **Amerikini** xuda saqlisun, **Amerikiniŋ** tibbi texnikisi ilğar, **Amerikiliqlar** bärdašliq beräläydu didi.	Word	Meaning	BM	BM-U
	almiliq bağ	apple orchard	N	Y
	almilar	apples	N	Y
	almaŋ	your apple	Y	Y
	almida	in(on) the apples	N	Y
	alma	apple	Y	Y
The apples in this apple orchard grow very well. Your apple is delicious. Apples are rich in vitamins. Apple is a symbol of Apple Inc. of the America. My brother mailed me an American-made iPhone from the America. My American friend said that the number of infected people in the America has exceeded 500,000, god bless the America, the America has developed medicine, and the American people can overcome the epidemic.	amerika	America	Y	Y
	amerikidin	from America	N	Y
	amerikida yasalğan	made in America	N	Y
	amerikiliq	American	N	Y
	amerikidiki	In the America	N	Y
	amerikini	(the) America	N	Y
	amerikiniŋ	the America's	N	Y
	amerikiliqlar	American people	N	Y

The search engines with the most users in China are Baidu (baidu.com) and Sogou (sogou.com). When these two search engines search for Uyghur words, from a technical point of view, the Uyghur word of length n is regarded as a Chinese word composed of n Chinese characters, so the accuracy of the search results is very low. When *P =*" ئالما" (Alma), the precision of the first 20 search results are P *(sougou) = 0.2, P (baidu) = 0.05*. The search results of Microsoft Bing (bing.cn) conform to the matching principle of BM algorithm, the precision of the first 20 search results is *P (bing) = 0.85,* and the search results do not include weakened vowel words. China's largest social platform Wechat and some authoritative Uyghur websites that can provide content retrieval services, such as business information network (uqur.cn) and Kunlun network (uyghur. xjkunlun. gov.cn) have similar search performance.

6.4. Monosyllabic and Non-syllabic Retrieval

Because there is no vowel weakening phenomenon of single syllables (independent syllables, not monosyllable words) and non-syllable strings, there is no difference in the retrieval results of the BM algorithm and the BM-U algorithm.

1. Monosyllabic retrieval

Retrieving single syllables in Tz is very convenient, and the two algorithms are equally efficient. When *P = Sb, Pz = Sc.* As shown in Table 10, *P = "ma"* and *P = "to"* are the search results of single syllables. The number of structural matches in *T* far exceeds the number of matches in *Tz*. A search in *T* is equivalent to a fuzzy match *P =% Sb%*, and a search in *Tz* is equivalent to an exact match. The search results of *T* include other syllables such as *or+man*, *mal*, *toğ+ra*, and *top*. If accurate monosyllabic retrieval is implemented in *T*, it will increase the technical difficulty and extra time consumption, because after finding a match, the string needs its syllables segmented, and then it is determined whether the match is an independent syllable and not a part of other syllables. Implementing fuzzy matching of single syllables in *Tz* also increases technical difficulty and time consumption because this requires each syllable in *Tz* to be decoded and then for fuzzy matching to be executed.

2. Non-syllable retrieval

Tz is syllable-encoded text. Since the basic unit of data storage is the syllable, it is impossible to retrieve non-syllable content. For example, the search result of *P = "mm"* in Table 10 is one because

there is an abbreviation *mm* in the text that meets the matching conditions. Searching with *T* is very convenient: we can just search directly.

Table 10. *T* and *Tz* monosyllabic and non-syllable string search comparison.

String Type	P	T (M. num)	Tz (M. num)	Mis. e.g.,
Monosyllable	ma	7053	1902	Or + **ma**n, **ma**l
	to	2718	1410	**To**ğ + ra, **to**p
Non-syllable	m, u, tt, mm, uu, ää	39455, 37998, 2802, 686, 19, 9	5, 14, 0, 0, 0, 0	

6.5. Comparison with Other Related Studies

1. Dawut [19] designed two functions *Bohum_Sani* (number of syllables) and *Bohum_Xekli* (syllable type) to propose a multi-pattern matching algorithm Bohum-Ug, which first applied syllables to pattern matching research. The algorithm first splits the syllables of pattern *P* and text *T*. When pattern matching, use *Bohum_sani* function to compare the number of syllables. If the number of syllables is the same, use *Bohum_Xekli* to compare the types of syllables. Then compare the characters after the same syllable types. This algorithm requires syllable segmentation in advance. When the size of the text *T* is large, the syllable segmentation consumes additional algorithm time. The final matching result is similar to the BM algorithm and cannot match weakened words. The BM-U algorithm does not require syllable segmentation, and can match weakened words, because the matching mechanism of the BM algorithm is not changed, and the BM-U algorithm can be transplanted to all variants of the BM algorithm.

2. Tohti [20] proposed WM-Uy (Wu-Manber-Uy), a multi-pattern matching algorithm. Stem extraction is performed on the pattern *P* before pattern matching. After the pattern matching of the stem is successful, the word suffix is checked. If the suffix is a derivational suffix, the matching fails, and when the suffix is an inflectional suffix, the matching is successful. The WM-Uy algorithm is different from the single-mode matching BM-U algorithm proposed in this paper. (1) The WM-Uy algorithm does not use stemming to match monosyllable weakened words, such as *P = {Eti (his horse), Eri (his man), Eqi (the white)}* cannot match the corresponding unweakened words *W = {At, Ar, Ak}*. (2) According to the Aizimaiti [48] WM-Uy algorithm suffixes library should include all 378 Uyghur suffixes (104 derivational suffixes, 274 inflectional suffixes). The BM-U algorithm does not have an suffixes library, for nouns compare weakened words up to four times. (3) The matching requirements of the WM algorithm are different from the BM-U algorithm. According to the requirements of the WM-Uy algorithm, when *P = {Alma, Amerika}*, the stem add derivational suffix words *W = {Almizar and Almiliq (Apple Orchard), Amerikiliq (American), Almimu (Apple is also ..., is it Apple?), Amerikimu, Almiči (the person who deals with Apple), Almixan (taking Apple as the female name)}* are not in the match, but in the BM-U algorithm, these words can satisfy the weakening forms of pattern *P* and can be matched. (4) The WM-Uy algorithm can match*Almas (diamond), Almax (exchange)* and other words that can match *P* in structure but are not semantically related to *Alma*. This paper proposes a syllable-based searchable compressed text format *Tz*; when *Tz* format cooperates with BM-U it can exclude these semantically unrelated words.

7. Conclusions

Uyghur is a very typical phonetic language. Each word is composed of syllables. The pronunciation of characters and syllables is the same as that of words. The *Tz* format proposed in this paper is a searchable compressed text format based on syllable encoding. The original document doc (char) with the character as the basic unit is changed to the document doc (Sb) with the syllable as the basic unit. If the *Tz* format is used as an auxiliary storage format for a text corpus, then based on the average length of a Uyghur syllable being 2.4 characters, the theoretical matching speed is 2.4 times faster when matching with a brute force algorithm. The *Tz* format is more convenient for accurate retrieval and processing of natural language content in units of syllables, requires less space, and matches faster.

It can exclude some semantically unrelated words without semantic analysis and requires a syllable encoding dictionary installed on the client. The *Tz* format design ideas can be used in other languages that can be segmented into syllables and have complex word form features [49]. Figure 3 shows the process of retrieving a text corpus using speech. vSb in the figure is the speech syllable corresponding to text syllable *Sb*.

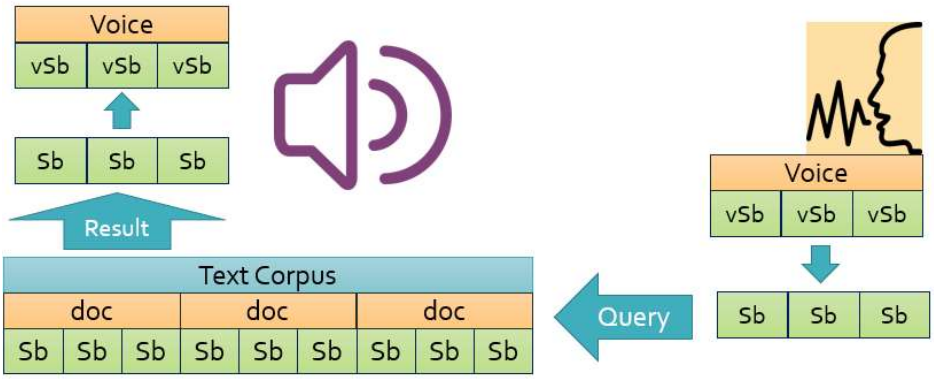

Figure 3. Principle of speech retrieval based on syllable-coded document corpus.

The BM-U algorithm proposed in this paper is designed on the basis of the original BM algorithm to address the complex morphology of Uyghur words. The retrieval object is Uyghur natural language content. The new design does not change the original search mechanism of the BM algorithm and upgrades the original matching method to an extended matching method based on *P_list*, where *P_list* can be calculated based on pattern *P*; thus, this improvement can be transplanted to other versions of the BM algorithm. This paper only considers the relationship between the stem and the 1-level syllables attached to the stem when designing *P_list*, which is a syllable-based unigram method. If the content of *P_list* is increased to the 2-level or 3-level syllables attached to the stem, it will become a syllable-based bigram and trigram problem. Increasing from unigram to bigram and trigram will increase the time consumption and technical difficulty but will help improve the precision and accuracy values. This extended matching idea can also be theoretically applied to multi-pattern matching methods such as Wu–Manber.

This paper mainly studies the pattern matching of nouns. Uyghur verbs have more suffix types and numbers than nouns, and their combination levels, structural changes, and additional rules are more complicated. When designing a verb generator algorithm, in theory, the number of forms based on a verb stem may reach thousands. The BM-U algorithm proposed in this paper requires mode P to be a stem. Uyghur stemming itself is one of the most important basic research contents, among which the stemming of verbs is more difficult. This study also found that spelling errors also have a certain effect on the efficiency of pattern matching. These are our future research directions.

Author Contributions: Conceptualization, W.A.; Formal analysis, W.A. and M.M.; Funding acquisition, A.W.; Investigation, J.W.; Methodology, W.A. and M.M.; Resources, J.W. and K.A.; Software, W.A., M.M. and H.W.; Writing—original draft, W.A.; Writing—review and editing, M.M., H.W., T.Y. and A.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Laboratory of Xinjiang Uyghur Autonomous Region of China, grant number 2018D04019; the National Natural Science Foundation of China, grant numbers 61762084, 61662077, 61462083; the Scientific Research Program of the State Language Commission of China, grant number ZDI135-54; and the National Key Research and Development Project of China, grant number 2017YFB1002103.

Acknowledgments: This work was supported by the Opening Foundation of the Key Laboratory of Xinjiang Uyghur Autonomous Region of China (grant number 2018D04019); the National Natural Science Foundation of China (grant numbers 61762084, 61662077, 61462083); the Scientific Research Program of the State Language Commission of China (grant number ZDI135-54); and the National Key Research and Development Project of China (grant number 2017YFB1002103).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ma, Z.F.; Yang, S.Y.; Guo, G.F. A fast improved pattern matching algorithm based on BM. *Control Decis.* **2013**, *28*, 1855–1858.
2. Zhou, X.; Xu, B.; Qi, Y.; Li, J. MRSI: A Fast Pattern Matching Algorithm for Anti-virus Applications. In Proceedings of the International Conference on Networking, Cancun, Mexico, 13–18 April 2008; pp. 256–261.
3. Du, F. A Faster Pattern Matching Algorithm for Intrusion Detection. *Adv. Mater. Res.* **2012**, *532*, 1414–1418. [CrossRef]
4. Tahir, M.; Sardaraz, M.; Ikram, A.A. EPMA: Efficient pattern matching algorithm for DNA sequences. *Expert Syst. Appl.* **2017**, *80*, 162–170. [CrossRef]
5. Gagie, T.; Gawrychowski, P.; Puglisi, S.J. Approximate pattern matching in LZ77-compressed texts. *J. Discret. Algorithms* **2015**, *32*, 64–68. [CrossRef]
6. Ablez, W.; Ayzohra; Niyaz, A.; Osman, I. Study on the Some Key Technology of Improving the Quality of Uyghur Search. *Math. Pract. Theory* **2013**, *43*, 119–123.
7. Xue, P.Q.; Xian, Y.; Nurbol; Silamu, W. Sensitive information filtering algorithm based on Uyghur text information network research. *Comput. Eng. Appl.* **2018**, *54*, 236–241.
8. Mahmoud, A.; Yusuf, H.; Jiajun, Z.H.A.N.G.; Chengqing, Z.O.N.G.; Hamdulla, A. Name recognition in the Uyghur language based on fuzzy matching and syllable-character conversion. *J. Tsinghua Univ.* **2017**, *57*, 188–196.
9. Kahaerjiang, A.; Tuergen, Y.; Tianfang, Y.; Aishan, W.; Aishan, M. An Improved Method for Uyghur Sentence Similarity Computation. *J. Chin. Inf. Process.* **2011**, *25*, 50–53.
10. Boyer, R.S.; Moore, S.J. A Fast String Searching Algorithm. *Commun. Acm* **1977**, *20*, 762–772. [CrossRef]
11. Wu, S.; Manber, U. *A Fast Algorithm for Multi-Pattern Searching*; Technical Report TR-94-17; University of Arizona: Tucson, AZ, USA, 1994.
12. Xiaohua, L. A Boyer-Moore Type String Matching Algorithm with Memory and Its Computational Complexity. *J. Hunan Univ. Nat. Sci.* **2008**, *35*, 84–88.
13. Yipe, W. An Improved Wu-Manber Multi-pattern Matching Algorithm for Chinese Encoding. *J. Chin. Comput. Syst.* **2015**, *36*, 778–781.
14. Nurmemet, Y.; Wushour, S.; Reyiman, T. Syllable based language model for large vocabulary continuous speech recognition of Uyghur. *J. Tsinghua Univ. Sci. Technol.* **2013**, *53*, 741–744.
15. Mamateli, T. Context dependent syllable based speech synthesis system for Uyghur. *Comput. Eng. Appl.* **2011**, *47*, 141–143.
16. Mahmut, M.; Turgun, I. A Research on Syllable Based Uyghur Text Proofreading System. In Proceedings of the the Ninth National Conference on Computational Linguistics, CCL 2007, Dalian, China, 6–8 August 2007.
17. Ranagul, D.; Askar, H.; Dilmurat, T. Acoustic Analysis on Prosodic Feature of CVC Type Syllable in Uyghur Language. *Comput. Eng.* **2011**, *37*, 193–195.
18. Isabel, R.M.C. Comparison of Uyghur and Spanish syllables. *China Natl. Exhib.* **2018**, *8*, 119–121.
19. Dawut, Y.; Abdureyim, H.; Yang, N.N. Research on Multiple Pattern Matching Algorithm for Uyghur. *Comput. Eng.* **2015**, *41*, 143–148.
20. Tohti, T.; Huang, J.; Hamdulla, A.; Tan, X. Text Filtering through Multi-Pattern Matching: A Case Study of Wu–Manber–Uy on the Language of Uyghur. *Inf. Int. Interdiscip. J.* **2019**, *10*, 246. [CrossRef]
21. Culpepper, J.S.; Moffat, A. Phrase-Based Pattern Matching in Compressed Text. In *International Symposium on String Processing and Information Retrieval*; Springer: Berlin, Heidelberg, 2006; Volume 4209, pp. 337–345.
22. Huynh, T.N.; Hon, W.K.; Lam, T.W.; Sung, W.K. Approximate string matching using compressed suffix arrays. *Theor. Comput. Sci.* **2006**, *352*, 240–249. [CrossRef]
23. YongKang, X.; Guanglu, Y.; Songfeng, L. Approximate string matching algorithm based on compressed suffix array. *Comput. Eng. Appl.* **2015**, *51*, 139–142.
24. Buluş, H.N.; Carus, A.; Mesut, A. A new word-based compression model allowing compressed pattern matching. *Turk. J. Electr. Eng. Comput. Sci.* **2017**, *25*, 3607–3622. [CrossRef]
25. Karkkainen, J.; Navarro, G.; Ukkonen, E. Approximate string matching on Ziv-Lempel compressed text. *J. Discret. Algorithms* **2003**, *1*, 313–338. [CrossRef]
26. Wang, J.F.; Li, Z.R.; Cai, C.Z.; Chen, Y.Z. Assessment of approximate string matching in a biomedical text retrieval problem. *Comput. Biol. Med.* **2005**, *35*, 717–724. [CrossRef]

27. Navarro, G.; Tarhio, J. LZgrep: A Boyer–Moore string matching tool for Ziv–Lempel compressed text. *Softw. Pract. Exp.* **2005**, *35*, 1107–1130. [CrossRef]
28. Quanzhu, Y.; Xiaojian, D.; Xueli, R.; Zhifeng, Z. Research of BWT-Boyer-Moore Compressed Domain Search Algorithm. *Appl. Res. Comput.* **2006**, *23*, 59–61.
29. Narupiyakul, L.; Thomas, C.; Cercone, N.; Sirinaovakul, B. Thai Syllable-Based Information Extraction Using Hidden Markov Models. In Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2004, Seoul, Korea, 15–21 February 2004; pp. 537–546.
30. Hackett, P.G.; Oard, D.W. Comparison of word-based and syllable-based retrieval for Tibetan (poster session). In Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages, Hong Kong, China, 30 September–1 October 2000.
31. Oflazer, K.; Kuruoz, I. Tagging and Morphological Disambiguation of Turkish Text. In Proceedings of the Conference on Applied Natural Language Processing, Stuttgart, Germany, 13–15 October 1994; pp. 144–149.
32. Hakkanitur, D.; Oflazer, K.; Tur, G. Statistical Morphological Disambiguation for Agglutinative Languages. *Comput. Humanit.* **2002**, *36*, 381–410. [CrossRef]
33. Xu, J.; Pan, J.; Yan, Y. Agglutinative Language Speech Recognition Using Automatic Allophone Deriving. *Chin. J. Electron.* **2016**, *25*, 328–333. [CrossRef]
34. Park, H.; Oh, K.; Choi, H.; Gweon, G. Constructing a paraphrase database for agglutinative languages. *Data Knowl. Eng.* **2019**, *123*, 101604. [CrossRef]
35. Saimaiti, A.; Wang, L.; Yibulayin, T. Learning Subword Embedding to Improve Uyghur Named-Entity Recognition. *Inf. Int. Interdiscip. J.* **2019**, *10*, 139. [CrossRef]
36. Mireguli, A.; Mijiti, A.; Aisikaer, A. A Morphological Analysis Based Algorithm for Uyghur Vowel Weakening Identification. *J. Chin. Inf. Process.* **2008**, *22*, 43–47.
37. Dawel, A.; Hayrat, H. Study on the Rule-based Kazakh Word Lemmatization Algorithm. *J. Xinjiang Univ. Nat. Sci. Ed.* **2011**, *28*, 116–119.
38. Saren, B. Research on the Causes of the Weakening and Even Disappearance of Short Vowels in Mongolian. *J. Inn. Mong. Univ. Natl. Soc. Sci.* **2005**, *31*, 29–31.
39. Hayes, B.; Siptar, P.; Zuraw, K.; Londe, Z. Natural and Unnatural Constraints in Hungarian Vowel Harmony. *Language* **2009**, *85*, 822–863. [CrossRef]
40. Goldsmith, J.; Riggle, J. Information theoretic approaches to phonological structure: The case of Finnish vowel harmony. *Nat. Lang. Linguist. Theory* **2012**, *30*, 859–896. [CrossRef]
41. Genxiong, J. The Weakening and Dropping of Vowels in Mongolian Language. *J. Inn. Mong. Univ. Natl. Soc. Sci.* **2010**, *36*, 27–29.
42. Qingxia, D.; Ling, W. Experimental Study on the Characteristics of the Weakened Syllables of Jingpo. *J. Minzu Univ. China Philos. Soc. Sci. Ed.* **2014**, *5*, 154–159.
43. Jaworski, S. Phonetic and Phonological Vowel Reduction in Russian. *Pozn. Stud. Contemp. Linguist.* **2010**, *46*, 51–68. [CrossRef]
44. Bakovic, E. Vowel harmony and stem identity. *San Diego Linguistic Papers.* **2003**, *1*, 1–42.
45. Xinjiang Uygur Autonomous Region National Language Working Committee. *Dictionary of Modern Uyghur Literature Language Orthography*; Xinjiang People's Publishing House: Urumqi, China, 1997.
46. Wayit, A.; Jamila, W.; Turgun, I. Modern Uyghur automatic syllable segmentation method and its implementation. *China Sci.* **2015**, *10*, 957–961.
47. Abliz, W.; Wu, H.; Maimaiti, M.; Wushouer, J.; Abiderexiti, K.; Yibulayin, T.; Wumaier, A. A Syllable-Based Technique for Uyghur Text Compression. *Inf. Int. Interdiscip. J.* **2020**, *11*, 172. [CrossRef]
48. Ainiwaer, A.; Jun, D.; Xiao, L.I. Rules and Algorithms for Uyghur Affix Variant Collocation. *J. Chin. Inf. Process.* **2018**, *32*, 27–33.
49. Tuergen, I.; Kahaerjiang, A.; Aishan, W.; Maihemuti, M. A Survey of Central Asian Language Processing. *J. Chin. Inf. Process.* **2018**, *32*, 1–13+21.