information

MDPI

Article

Text Filtering through Multi-Pattern Matching: A Case Study of Wu–Manber–Uy on the Language of Uyghur

Turdi Tohti [1], Jimmy Huang [2], Askar Hamdulla [1,*] and Xing Tan [2]

[1] College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China
[2] Information Retrieval & Knowledge Management Research Lab, York University, Toronto, ON M3J 1P3, Canada
* Correspondence: askar@xju.edu.cn

Received: 14 May 2019; Accepted: 19 July 2019; Published: 24 July 2019

Abstract: Given its generality in applications and its high time-efficiency on big data-sets, in recent years, the technique of text filtering through pattern matching has been attracting increasing attention from the field of information retrieval and Natural language Processing (NLP) research communities at large. That being the case, however, it has yet to be seen how this technique and its algorithms, (e.g., Wu–Manber, which is also considered in this paper) can be applied and adopted properly and effectively to Uyghur, a low-resource language that is mostly spoken by the ethnic Uyghur group with a population of more than eleven-million in Xinjiang, China. We observe that technically, the challenge is mainly caused by two factors: (1) Vowel weakening and (2) mismatching in semantics between affixes and stems. Accordingly, in this paper, we propose Wu–Manber–Uy, a variant of an improvement to Wu–Manber, dedicated particularly for working on the Uyghur language. Wu–Manber–Uy implements a stem deformation-based pattern expansion strategy, specifically for reducing the mismatching of patterns caused by vowel weakening and spelling errors. A two-way strategy that applies invigilation and control on the change of lexical meaning of stems during word-building is also used in Wu–Manber–Uy. Extra consideration with respect to Word2vec and the dictionary are incorporated into the system for processing Uyghur. The experimental results we have obtained consistently demonstrate the high performance of Wu–Manber–Uy.

Keywords: Uyghur; multi-pattern matching; Wu–Manber; Wu–Manber–Uy; text filtering

1. Introduction and Motivation

Living in an era of big-data, we are endowed with unprecedentedly abundant opportunities to access information and knowledge that are also very large in content and size. Unfortunately, more and more often, big data and big information are mixed with erroneous and mistrustful noises. A booming growth in social media in recent years in particular, calls for development and progress in techniques and tools for filtering out textual noises [1]. Regarding that aspect, content-based filters and models are particularly successful [2], and the ones for the English language have already been widely used in various fields, including information retrieval [3,4], network security [5,6], personalized information extraction and inference [7], content-based recommendation systems [8], knowledge discovery [9,10], Short Message Service (SMS) spam filtering [11,12], and many other areas.

It is the case that, when dealing with voluminous and complex so-called big data, traditional approaches and technologies in data management and text processing are inadequate in their functionalities and inefficient in performance, stirring a surge in research (in various specific subjects including resource sharing, knowledge acquirement, and content security) for foundational progress

and practical improvements on the problem to find better solutions. Not only restricted to major languages in the world such as English or Chinese, the need to deal with Uyghur in big data is particularly urgent in the following three senses, as we see them. (1) With the rapid development of large-scale storage technology and services in internet and digital communications, massive amounts of digitalized text data in Uyghur have been created in an accelerated speed and on a day-to-day basis. (2) While the lawful use of information extracted from these massive text data should be encouraged and promoted, the existence and spread of harmful or unauthentic information poses serious and in many cases severe threats to regional security and stability in Xinjiang, China. (3) Though Uyghur itself belongs to the category of low-resource languages, it is also is an agglutinating language, and several unique linguistic features only exist in Uyghur.

Whether it is for information acquisition or for protection against unwanted information, from our discussion above, technologies and tools for text filtering in Uyghur are urgently needed, and, unfortunately at the same time, existing text filters only work for either English, Chinese, or Arabic. These cannot be directly applied for our purpose. Research reported in this paper aims to bridge this gap. In the expansion aspect, it combines the Uyghur–Chinese bidirectional dictionary and the deep-learning tool Word2vec [13], and it uses multi-pattern matching as the key mechanism adopted in our model. The model has demonstrated impressive filtering efficiency and effectiveness.

The main contributions of this paper are: (1) The inadequacy and deficiency of the multi-pattern matching algorithm Wu–Manber in Uyghur for the first time have been observed, leading to the development of Wu–Manber–Uy (Wu–Manber for Uyghur). The algorithm not only solves the mismatch problem caused by morphological characteristics but also reduces the negative impact that spelling mistakes pose on the algorithm. (2) We propose a model expansion strategy which is a combination of a dictionary and Word2vec. The strategy is applied in our Uyghur text filtering system is based on multi-pattern matching. The effectiveness of the algorithm is tested and verified on an actually implemented system. (3) The research and work presented in this paper can be generalized to perform text filtering tasks in a series of low-resource languages.

The remainder of the paper is organized as follows. Section 2 introduces research works related to this paper. In Section 3, we apply Wu–Manber to Uyghur to show its limitations and focus on our efforts to the revision of Wu–Manber, that is, Wu–Manber–Uy (Wu–Manber for Uyghur). This Section also introduces the basic framework and the model for the actual application of Wu–Manber–Uy in Uyghur text filtering. In Section 4, experiments in comparison between Wu–Manber and Wu–Manber–Uy are performed on the over-all efficiency of text filtering. A further analysis and discussion of the experimental results are also covered in this section. Finally, Section 5 concludes the paper.

2. Related Work

Compared with Chinese, English, or other major languages, Uyghur was considered late in research related to the subject of text filtering. We observed that under the quickly surging demand for the availability of text filtering in Uyghur, more than a few preliminary results were obtained to this end. For example, in a paper by Almas et al. [14], after some pre-processing steps, similarities between text and filter templates were measured with respect to the commonly used similarity metrics. Thus, the filtering efficiency for Uyghur was investigated. This was in contrast to Zhao et al. [15], who proposed an integrated approach taking both document frequency and mutual-information into consideration to select features for building up text models; filtering was consequently performed.

Different from those approaches that were based on classification and information retrieval, pattern matching-based methods focus on the process of finding specific patterns from texts. One of the key features of pattern matching-based methods is: In such a method, one does not need to construct any specific text model, any marking, or any training on data beforehand [16]. These methods have achieved a notably great performance in terms of time efficiency—being able to handle the task of content filtering for large-scale texts. Research on content filtering based on pattern matching is

currently very active, and many working algorithms, models, and frameworks have been proposed over the past years, as pointed out in [17,18].

Research on text filtering in Uyghur based on pattern matching has started in recent years. For example, Dawut et al. [19] proposed a multi-pattern matching algorithm based on syllable division in Uyghur. The algorithm has been used for text filtering. This method requires the syllable decomposition of texts and the availability of patterns in advance. Additionally, special grammatical features must also be considered. Therefore, the time efficiency of the method cannot meet the requirements for large-scale text filtering. In addition, in a paper by Xue et al. [20], decision trees were used to represent user patterns (sensitive words) to perform word-by-word scanning and the matching of texts. This method accelerates the scanning speed to some extent, but the cost of time in constructing and updating the decision tree cannot be neglected either.

Pattern matching is also widely used in other text mining tasks in addition to text filtering, such as knowledge extraction [21,22] (named entity recognition through pattern matching [23,24]) or question classification/answer extractions in question-answering systems [25–27]. Recent research has investigated the applicability of this technique for Uyghur semantic entity extraction, personal name recognition, and other research. For example, Tohti et al. [28] proposed a method for rapid semantic string extraction in Uyghur through a statistical and shallow language parsing approach: First, an improved n-gram incremental algorithm is used for word string expansion [29]; second, they search for reliable frequent textual patterns whose structural integrities are scrutinized; and finally, all semantic strings in the text are found. In a paper by Muhammad et al. [30], Uyghur person name and pattern libraries were established separately. In their proposed method, names from knowledge bases, together with pattern matching methods, are used to allocate Uyghur person names in the text. However, the method was not based on fast multi-pattern matching. Additionally, Yusuf et al. [31] proposed a letter-based fuzzy matching recognition method to identify the person name in texts according to the matching degree between the name under investigation and the names collected in the knowledge base. It needs to be remarked, however, that all text in entirety needs to be scanned sequentially for this method.

In our research, we investigated how the famous Wu–Manber algorithm can be adjusted and applied to do the multi-pattern matching tasks in Uyghur. We accordingly propose Wu–Manber–Uy (standing for Wu–Manber for Uyghur), which is a significant improvement to Wu–Manber in the context of content-based text filtering in Uyghur. The idea for the revision and improvement was based on a comprehensive understanding of the Uyghur language itself and on how Uyghur linguistically differs with English. We have built a text filtering system for Uyghur, centred at Wu–Manber–Uy. In the system, dictionary and Word2vec-based approaches are further considered for pattern expansion. Our obtained experiments results consistently demonstrate the high performance of Wu–Manber–Uy and the system in general.

3. Our Proposed Method for Uyghur Multi-Pattern Matching and Text Filtering

In this section, we discuss the applicability and efficiency of the Wu–Manber algorithm on Uyghur. We took measures on the issues and problems we have observed and encountered in both the preprocessing and matching stages of the algorithm. Correspondingly, we developed Wu–Manber–Uy, a multi-pattern matching algorithm suitable for Uyghur. In Section 3.3, we propose a text filtering system for Uyghur, employing Wu–Manber–Uy as its core algorithm.

3.1. The Deficiency of Wu–Manber Algorithm in Uyghur

In the scanning process, the Wu–Manber algorithm takes "block" as the unit to look for patterns in the text. A "block" is a substring of size B (usually B is assigned to 2 or 3). The shift distance of the matching window is decided by B [32]. Pre-processing in the algorithm includes word segmentation in Chinese and stemming in Uyghur [33]. The Uyghur language belongs to an agglutinating (adhesive) language with abundant morphological and grammatical rules. In Uyghur, *word* is the smallest

language unit, and *stem* represents the lexical meaning of a word, so stemming is essential for determining patterns.

Wu–Manber builds three tables: SHIFT, HASH, and PREFIX. Taking an example consisting of three patterns handled by stemming: "ئالما" (apple), "مېۋە" (fruit) and "تېلېفون" (telephone), let $B = 2$, and the matching window size $m = 4$. To build the table SHIFT, we take the first m characters of each pattern into account and calculate the distance between the tail of the pattern and the block in the matching window, which equals to the shift distance of the matching window when the block occurs in the text. Blocks and corresponding shift distance for these three patterns are shown in Figure 1 below.

ما	لم	ئال
0	1	2

: ئالما

ۋە	ېۋ	مې
0	1	2

: مېۋە

لې	ېل	تې
0	1	2

: تېلېفون

Figure 1. Blocks and shift distances.

The results are combined to construct the following table for SHIFT in Figure 2. All other blocks that do not occur in the matching window have the same shift distance of 3 (i.e., $m - B + 1$). The values in the SHIFT table mark how far one can shift forward (skip) if these blocks occur in the scanning text.

...	لې	ېل	تې	ۋە	ېۋ	مې	ما	لم	ئال
3	0	1	2	0	1	2	0	1	2

Figure 2. SHIFT table.

The HASH table is further built, as shown in Figure 3. Each entry of the HASH table (for example, HASH[i]) points to a list of patterns whose last B characters are hashed into the location.

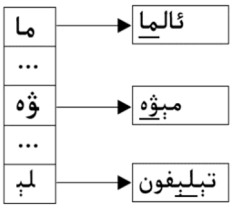

Figure 3. HASH table.

The third step is to build the PREFIX table, and the result is shown in Figure 4. For any SHIFT value of 0, the HASH table is consulted in order to decide whether or not there is a match. We check specifically the values in the PREFIX table to filter out patterns with exactly the same suffix but with different prefix.

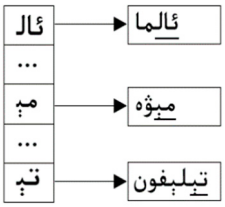

Figure 4. PREFIX table.

In the Wu–Manber algorithm, pre-processing can be performed in a quick time, leading to the subsequent scan and matching stage.

Using the Wu–Manber algorithm and parameter values as set in the previous section, the scanning and matching of a string in Uyghur is illustrated in Figure 5 (Uyghur writes and reads from right to left, and the English meaning of the sentence below is: Smelling apples can help relieve migraines).

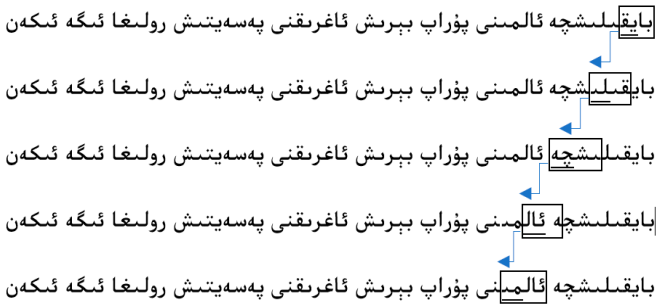

Figure 5. Word-based string scanning.

From this matching process, we spotted the following problems, which, as we note, are caused by unique linguistic features of Uyghur.

Problem 1: Though pattern "ئالما" appears in the above text, the pattern is subject to a third person accusative deformation, since after the word-forming affix "نى" occurs before the pattern, leading to an occurrence of a mismatch. In Uyghur, this phenomenon is known as vowel weakening [34]. The letters "ە" or "ا" in the last syllable of a stem should be replaced by the letters "ئى" or "ئې" with some affix attached to them. An example of vowel weakening is specifically shown in Figure 6.

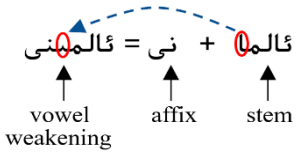

Figure 6. An example of vowel weakening.

It appeared to be the case that we could solve the problem of vowel weakening by simply using the stemming approach. Nonetheless, doing so would arise a new problem in the string scanning process, as shown in Figure 7 (the English meaning of the sentence below is: I bought a house in the apple orchard community).

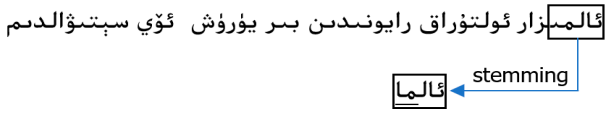

Figure 7. Stem-based string scanning.

Problem 2: The word "ئالمىزار(apple orchard) in the string is a new word constructed by the stem "ئالما" and the word-building affix "زار," which, when bind together as "ئالما +زار =ئالمىزار," refers to a place. If it is stemmed, it will match the pattern "ئالما," but this is wrong [35].

In observation of the above challenges and difficulties we have been facing when customizing Wu–Manber for Uyghur, we propose Wu–Manber–Uy, a variant to Wu–Manber that is dedicated to work for filtering texts in Uyghur.

3.2. Our Multi-Pattern Matching Algorithm: Wu–Manber–Uy

For mismatching caused by vowel weakening, we can do stemming to the text and all patterns first, and then we can conduct stem-based scanning and matching afterwards. However, computationally, this is costly and cannot satisfy the requirement of fast pattern matching in dynamic settings for text flow filtering [36]. Therefore, instead of doing stemming, we looked at approaching the problem differently.

An outstanding merit of Wu–Manber, as we know, is that the algorithm uses blocks of size B to extend the effect of shifting for the patterns not appearing in the string, and it uses a HASH table to speed up the filtering of the patterns that should be matched during the matching stage. Variances on the number of patterns do not affect performance of the algorithm a lot unless the number of patterns is increased to a few thousand, which will then cause a notable jump in the need of time for computing matching [37], granting us an opportunity for improvement: If the last syllable of a stem of a pattern has the letter "ئە" or "ئا," the deformation of this pattern will be added as a new pattern in consideration of vowel weakening. For example, as for the stem "ئالما," there is a deformation "ئالمى," so there are several patterns in the pattern-set in the example; that is, a pattern set of {"ئالما," "ئالمى,"" مېۋە," and "تېلېفون"}. The SHIFT table and HASH table can be adjusted accordingly, as shown below in Figures 8 and 9.

...	لېـ	ېلـ	تېـ	ۆە	ېۋ	مېـ	مـ	ما	لم	ئال
3	0	1	2	0	1	2	0	0	1	2

Figure 8. SHIFT table for Wu–Manber–Uy.

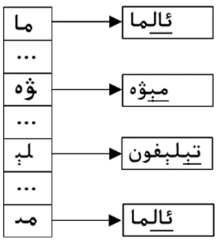

Figure 9. HASH table for Wu–Manber–Uy.

To deal with the issues raised in Problem 2, we sought solutions by considering the morphological features of Uyghur words which can be divided into roots, stems, and affixes. Among them, affixes are morphemes used to derive new words. They are attached to the root or stem and can be further divided into word-forming affixes (morphological affixes) and word-building affixes according to their functions.

A word-forming affix makes a morphological change of a word, with the lexical meaning of the word kept unchanged. In the following example, the word-forming affixes "نى" and "نىڭ" do not change the lexical meaning of the stem "ئالما."

(the apple)	ئالمنى	=	(word-forming affix)	نى	+	(apple)	ئالما
(of the apple)	ئالمنىڭ	=	(word-forming affix)	نىڭ	+	(apple)	ئالما

Differently, a word-building affix is used to construct a new word which has a completely new lexical meaning from the meaning of the stem. In the following example, the word-building affixes "زار" and "خان" completely change the lexical meaning of the stem "ئالما" and construct the new words "ئالمزار" and "ئالمخان."

(apple orchard)	ئالمزار	=	(word-building affix)	زار	+	(apple)	ئالما
(person name)	ئالمخان	=	(word-building affix)	خان	+	(apple)	ئالما

Therefore, when a pattern is matched with a word in the text, it is necessary to observe their affixes decide whether they actually match. Table 1 below shows example Uyghur affixes collected

and annotated in this work. In the table, *wf* refers to affix type of word-forming and *wb* refers to affix type of word-building.

Table 1. Set of Uyghur affixes.

Affix	Type
نىڭ	*wf*
دىن	*wf*
خان	*wb*
نى	*wf*
كار	*wb*
زار	*wb*
...	...

To recap, algorithmically, Wu–Manber–Uy extends and advances Wu–Manber, specifically in the following ways:

(1) It conducts the stemming process for all patterns first, which creates a pattern-set consisting of pattern stems and their deformations, and then builds the SHIFT table, the HASH table, and the PREFIX table:

(2) Matching and shifting start with the first word in the text (scanning is the same as the traditional Wu–Manber algorithm), If a match is found, go to (3), otherwise continue to skip according to the given distance in SHIFT table.

(3) Perform a stem extraction to the *i*'th word that matches in the text; we have stem W_i and its affix $W_{i\text{-}affix}$, the stem in the corresponding pattern P_i, and its affix $P_{i\text{-}affix}$; if W_i and P_i are both "zero-affix stem" (i.e., $W_{i\text{-}affix} = Null$ and $P_{i\text{-}affix} = Null$), the matching is marked as "correct;" otherwise, go to (4).

(4) If ($W_{i\text{-}affix} = Null$ and $P_{i\text{-}affix} \neq Null$ and $P_{i\text{-}affix} \in wf$), or ($W_{i\text{-}affix} \neq Null$ and $P_{i\text{-}affix} = Null$ and $W_{i\text{-}affix} \in wf$), or ($W_{i\text{-}affix} \neq Null$ and $P_{i\text{-}affix} \neq Null$ and $W_{i\text{-}affix} \in wf$ and $P_{i\text{-}affix} \in wf$), then this matching is also correct. Otherwise, go to (5).

(5) If the pattern and word are same (i.e., $W_{i\text{-}affix} = P_{i\text{-}affix}$), then this matching is correct. Otherwise, it is incorrect.

3.3. Our Proposed Method for Uyghur Text Filtering

In pattern matching-based text filtering, patterns are treated as "user profiles" to scan the text and to search for all matchings and related information. Given that, the same word in the pattern-set might have different meanings and the set is, to a considerable extent, limited in expressing semantics accurately, it is necessary to expand and refine the pattern-set through a dictionary. In this process, most words and phrases that are related to patterns are added to build up an extended pattern-set which is able to express the users' intention more accurately [38].

With recent progress in deep learning, meanings of words can be captured more comprehensively by computers and their AI programs in deep-learning. Significant improvements in word embedding, specifically in the field of synonym expansion, query expansion, inter-word semantic correlation calculation, and so on [39], have also been achieved in recent years. Our system also makes use of the technique of Word Embedding. A dictionary, as mentioned above, is also used to expand patterns in the system. Wu–Manber–Uy, as implemented to this end, has demonstrated efficient filtering ability working on text-sets in Uyghur. A flowchart of the Uyghur text filtering process is shown in Figure 10.

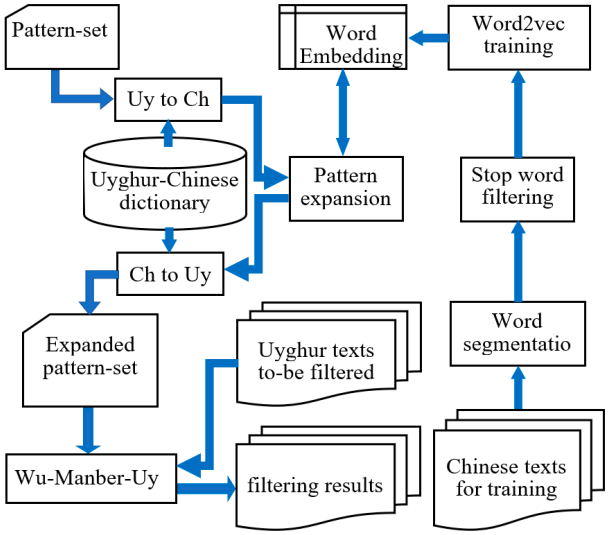

Figure 10. Flowchart of Uyghur text filtering process.

In this system, we first find the most similar words for each pattern using a Uyghur–Chinese dictionary and the deep learning tool Word2vec. Thus, new words can be added to the pattern-set. We then conduct pattern matching and filtering based on this expanded pattern-set. It needs to be remarked that Word2vec at its current stage is unable to directly handle Uyghur. Hence, here we used Chinese as an intermediate carrier between English and Uyghur for pattern expansion.

4. Experiments and Analysis

4.1. Data Sets

Computational linguistic research in Uyghur, specifically in text filtering, has just started. Thus far, there is no open standard dataset available. In order to carry out this current research, we actually collected and prepared the following data sets (freely available upon request for educational and academic research purposes):

(1) Chinese text-set: Word2vec training corpus in Chinese, collected from several official websites in Chinese, content related to politics, science and technology, and economics, with a size of 197.4 M.

(2) Uyghur text-set: Testing corpus in Uyghur, collected from several official websites in Uyghur, content related to politics, science and technology, and economics, with a size of 99.6 M.

In addition, Chinese–Uyghur bi-direction dictionary with a vocabulary of 418,533 items, developed by the key laboratory of multilingual information technology—Xinjiang, China.

4.2. Performance of Multi-Pattern Matching Algorithms

In this experiment, we compared performance in both time efficiency and matching accuracy between Wu–Manber in its original settings and Wu–Manber–Uy. Several benchmark sets of different sizes in pattern and in text were tested for the purpose. We compared the scanning and matching time (text preprocessing time was not included). All experiments were run on a computer of Core 2 Duo 3.0G CPU, 2.0G RAM. The operating system was Linux (ubuntu14.04). Programs were written in C++.

4.2.1. Comparison of Time Performance

For this comparison, we used a subset in the Uyghur text-set (with a size of 10.3 M). The patterns considered were Uyghur words (length ≥ five characters) randomly selected from the test-text, and the block length of B was 2. The running time of the two algorithms in comparison with the number of patterns increasing from 5 to up to 100 are shown in Figure 11.

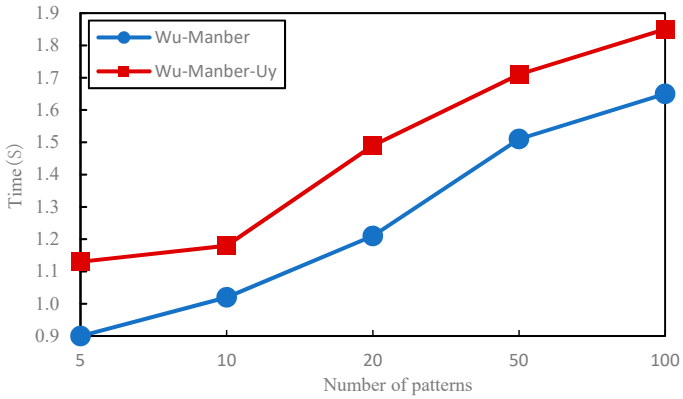

Figure 11. Comparison of time performance.

We can note from Figure 11 that Wu–Manber–Uy requires slightly more time than Wu–Manber. This is because, to solve the mismatch caused by vowel weakening, for each pattern in the pattern-set, Wu–Manber–Uy needs to additionally include two kinds of stem deformations in order to expand the pattern-set. In fact, a stem deformation should be added, but to determine if the letter "ئا" or "ئە" in the last syllable of a stem is replaced by "ئى" or "ئې," contextual analysis or stem library are needed. Hence, more time is needed here for Wu–Manber–Uy. In fact, for a given matching task, in the worst case, number of patterns in the Wu–Manber–Uy algorithm is three times more than the one in Wu–Manber. Wu–Manber–Uy, in addition, needs to match affixes for identified stem-based pattern matching, which takes more computation time.

4.2.2. Comparison of Matching Accuracy

With same experimental settings, we randomly selected ten patterns (length ≥ five characters) from the test-set. Meanwhile, the test-set was divided into three random corpora sizes of 19.3, 30.2, and 49.7 M. The matching accuracy of the two algorithms was compared with respect to these test sets, and the results are shown in Figure 12.

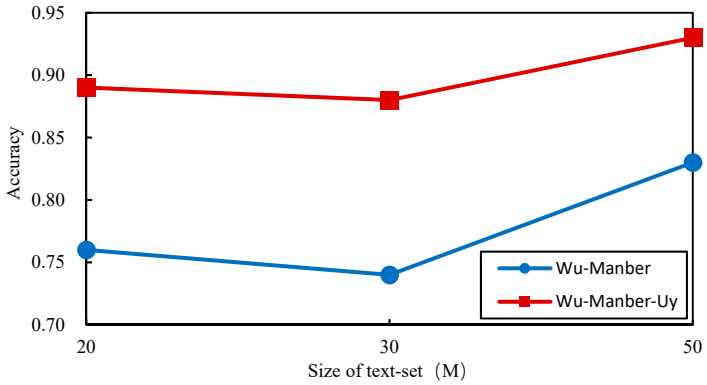

Figure 12. Comparison of matching accuracy.

In Figure 12, Wu–Manber–Uy has a significantly higher accuracy than Wu–Manber. This indicates that vowel weakening and affix-based matching, as considered in Wu–Manber–Uy, actually make a difference regarding filtering accuracy. Additionally, when we added stem deformations to the pattern-set, not only did they help solve the mismatch caused by vowel weakening, they also enabled the algorithm to bypass mismatches caused by spelling mistakes. In Uyghur, for example, "ئى" and "ئې" are two letters that can be easily misspelled as the other, and frequently people mistakenly spell "ئالمىنى" for "ئالمېنى."

4.3. Text Filtering Experiments

In this section, we mainly compare filtering precision and recall between Wu–Manber and Wu–Manber–Uy before and after pattern expansion. Specifically, we had (1) Chinese text-set for the Word2vec training corpus with a size of 197.4 M, segmented by the Stanford Chinese word segmentation tool; (2) we had an Uyghur text subset for the filtering test with a size of 49.7 M and a raw text corpus gathered from Uyghur Web portals.

In the preprocessing and pattern expansion stage, the top 10 words with the highest correlation degrees are selected and added to the pattern-set. The experimental results of matching and filtering which used the traditional Wu–Manber algorithm before and after pattern expansion are shown in Figure 13. The experimental results of the matching and filtering used by the Wu–Manber–Uy algorithm before and after pattern expansion are shown in Figure 14:

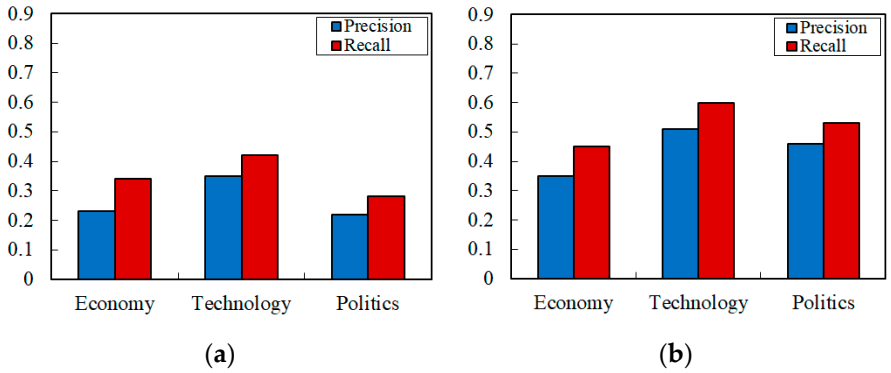

Figure 13. Filtering efficiency based on Wu–Manber multi-pattern matching (**a**) before pattern expansion and (**b**) after pattern expansion.

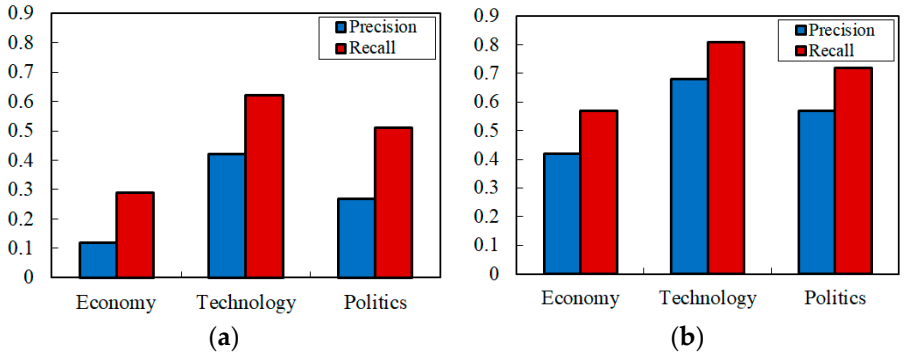

Figure 14. Filtering efficiency based on Wu–Manber–Uy multi-pattern matching (**a**) before pattern expansion and (**b**) after pattern expansion.

From these two figures, we can see that pattern expansion brings in noticeable improvement to both Wu–Manber and Wu–Manber–Uy, with Wu–Manber–Uy surpassing Wu–Manber. It should be noted that the filtering precision and recall on test-set of "Technology" were always higher than the other two test-sets. This is because the patterns selected for this experiment were more relevant to the "Technology" fields.

4.4. Further Discussion and Analysis

The above experimental results show that our method is effective. Here, we further analyze the merits of our proposed method and some factors that still somewhat affect its efficiency.

(1) In the related work of the past, stem segmentation has mostly taken up the workload of pre-processing in Uyghur. Meanwhile, text normalization and proofreading before stemming will

also consume a certain amount of time, and this especially is the case for large-scale text processing. The pattern matching method proposed in this paper, however, firstly divides the patterns into stems since stemming directly on the text can be largely avoided; the algorithm will only perform stemming on words until a match is found, saving the processing time for performing stemming on large-scale text corpus.

(2) Due to their heavy dependence on the availability of the stem library [33], current stem segmentation algorithms are not always able to correctly extract stems from infrequent words (i.e., words not included in stem libraries), leading to unfortunate cases where a word can be completely removed from the text. For example, for the Uyghur word "كەركىدانناڭ" (rhinoceros), the stemming algorithm first decomposes it morpheme, namely "كەر," "كى," "دان," "ناڭ." Since the stem "كەركىدان" (rhinoceros) is not found in the stem library but all its morpheme components are in the affixes library (here, we additionally observe an interesting morphological feature—a combination of Uyghur affixes also can construct a new Uyghur word), all these affixes are all filtered out, so the word will be removed from the text. The situation is worse when we are dealing with words new to the library. Consequently, some words that can provide important clues may be get dropped completely in the stemming process. Our current approach can effectively avoid happenings of this phenomenon.

(3) Uyghur is linguistically agglutinative. Several pairs of letters in its character set are similar to each other in appearance. Meanwhile, people from different regions might pronounce same words differently. Misspellings in Uyghur writing is thus quite common [40]. For large-scale texts, there is currently no method for proofreading the spelling in the batch [41]. Therefore, if the text processing method we adopt can directly or indirectly reduce the impact of spelling errors, it will bring about a considerable improvement for the entire system. Misspelling examples, as shown in Table 2 below, are drawn from actual articles in the test set (e.g., the letter "ئى" is incorrectly written as "ئي"). From a quick search on Bing (performed on Nov 11, 2018), one can easily confirm that such misspellings do exist on actual webpages.

Table 2. Misspelling examples and their actual occurrences on web pages (searched using Bing).

Stem	Correct Deformations	Documents Found	Misspelled Deformations	Documents Found
شىركەت(company)	شىركىتى	351,000	شىركېتى	83
خەتەر(danger)	خەتىرى	42,300	خەتېرى	77
مەشئەل(torch)	مەشئىلى	3660	مەشئېلى	97
مۇھابىبەت(love)	مۇھابىبىتى	2010	مۇھابىبېتى	72
ھۆكۈمەت(government)	ھۆكۈمىتى	208,000	ھۆكۈمېتى	272

In our method, however, for all erroneous and correct deformations to each stem, we created a SHIFT and a HASH table for them. In doing so, the matching algorithm is not affected by errors in spellings.

(4) In the articles of the test set, we also found nonstandard use of nouns, especially those new words translated directly from English. Though the uses of these words have been standardized by authorities, the arbitral use of them in writing or verbally are quite common [42]. For example, the standard term "كومپيوتېر" (computer) is abbreviated into "كوم," "ئېلېكترونلۇق يوللانما" (e-mail) is abbreviated into "ئېلخەت" or "ئېمايل" and "يۇمشاق دېتال" (e-mail) is written as "يۇمشاقدېتال" or "يۇمتال." In this work, we used the Uyghur–Chinese dictionary as an intermediate translation tool to find related words in the word vector model to expand the pattern-set. Because only canonical representations are presented in the dictionary and the word vector model is trained from a Chinese corpus, variants of one same word consequently cannot be found in the word vector model, non-surprisingly affecting the filtering efficiency.

4.5. Impact of the Proposed Methods

Multi-pattern matching is one of the a few leading methods in text content filtering. This method finds pattern-related or undesired text from the text set according to pattern given by users. It is fast and does not require the labeling of data of the construction of a special text model. Thus, the method is quite suitable for large-scale text filtering [43]. When combined with an adaptive learning mechanism [44] or aclassification or retrieval model [45,46], a better filtering efficiency can be obtained. In turn, filtering methods can also provide services for classification and retrieval.

In recent years, encouraging and promising results have been successfully obtained in the classification and retrieval of Uyghur [47,48]. The multi-pattern matching method proposed in this paper can be introduced into the Uyghur classification and retrieval system and take an active role. At the same time, it can also enrich the research content in the following areas, as we see them.

(1) For a low-resource language like Uyghur, a semi-supervised learning method should be the most suitable, because this method does not require a large amount of labelled corpus. In contrast, it uses bootstrapping to learn directly from large amounts of unlabeled text, assisted by "Seed" or "Sample," to guide the whole learning process [49]. In Uyghur-named entity recognition research, we used the multi-pattern matching method proposed in this paper to scan a small number of high-frequency entities or patterns (the ones such as common names and adjacent strings that often appear in a given context). We used them as seeds and extracted new ones. In doing so, the set of entities or patterns increased continuously, thus helping fast-named entity recognition.

(2) In the automatic question answering system, question classification and answer extraction are the two most important research topics, and a method based on deep learning is currently leading research [50,51]. Such research for Uyghur, however, is in its initial stage. Thus, it is necessary to build a question classification system and a basic resource library for the language in the first place. As such, we can establish a collection of patterns on the basis of the Uyghur Question Classification System and then use the pattern matching method proposed in this paper (with a possibly combination of additional machine learning techniques) to classify questions. The method is also, in a mainstream way, used for an answer extraction method, one particularly suitable for extracting answers to questions on definitions. The key aspect is to construct a set of answer patterns with a large coverage on various subjects [52,53]. A manually built model is costly and, in most cases, unrealistic. Nevertheless, we can solve this problem by pattern matching and pattern learning, as proposed in this paper.

(3) The template-based method is a mainstream machine translation method. In this method, the manually written translation template contains linguistic knowledge and can accurately capture the correspondence between language pairs. Though these templates are highly accurate, they do not have probabilistic information, are difficult to apply to statistical machine translation systems, and are prone to conflict when being matched [54]. The pattern matching method proposed in this paper can be applied to the statistical machine translation of Uyghur–Chinese and Uyghur–British text-sets. A high-quality translation template can be obtained through matching and learning. At the same time, the problem of template matching conflict can be solved.

5. Conclusions and Future Work

Wu–Manber is widely used for both Chinese and English multi-pattern matching due to its high performance in both time efficiency and accuracy in matching results. Nevertheless, the morphological characteristics of Uyghur make it difficult for Wu–Manber to keep its edge for the tasks of pattern matching in Uyghur. In specifically considering both stem deformation-based pattern expansion and the two-way matching based on the stem and affix, we proposed Wu–Manber–Uy, which is a revision and extension to Wu–Manber, for text filtering in Uyghur. We further built up a text filtering system based on Wu–Manber–Uy. In the system, patterns are expanded using a dictionary and the deep learning tool Word2vec, and a multi-pattern matching algorithm is used to scan and filter large-scale Uyghur text. Experimental and empirical analyses pervasively indicate that Wu–Manber–Uy surpasses Wu–Manber in Uyghur text filtering in aspects of both time efficiency and patching matching accuracy.

Using Wu–Manber–Uy for an actual deployment of text filtering system in Uyghur is suggested. There exists considerable space for further improvement. For example, although we have made extension to the pattern set, this extension, at its current stage, is largely static. Feedback mechanism can thus be further introduced into the system. Through this adaptive learning method, the filtering pattern will be updated dynamically during the filtering process. Over time, the pattern set will converge into one that better satisfies the filtering requirement of the users of the system. We will save this direction of research for our future work.

Author Contributions: Conceptualization, T.T. and A.H.; data curation, T.T. and A.H.; formal analysis, T.T. and J.H.; methodology, T.T.; experimentation, T.T.; writing—original draft, T.T.; writing—review & editing, J.H., and X.T.

Funding: This research has been supported by the National Natural Science Foundation of China (61562083, 61262062), and National Key Research and Development Plan of China (2017YFC0820603). This research is also supported in part by a discovery grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada. All the work was done when the first author was a visiting professor at the Information Retrieval and Knowledge Management Research Lab, York University, Canada.

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

References

1. Khurshid, S.; Khan, S.; Bashir, S. Text-Based Intelligent Content Filtering on Social Platforms. In Proceedings of the International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2014; pp. 232–237.
2. Bertino, E.; Ferrari, E.; Perego, A. A General Framework for Web Content Filtering. *World Wide Web-Internet Web Inf. Syst.* **2010**, *13*, 215–249. [CrossRef]
3. Renugadevi, S.; Geetha, T.V.; Murugavel, N. Information Retrieval Using Collaborative Filtering and Item Based Recommendation. *Adv. Nat. Appl. Sci.* **2015**, *9*, 344–349.
4. Wang, X.C.; Li, S.; Yang, M.Y.; Zhao, T.J. Personalized Search by Combining Long-term and Short-term User Interests. *J. Chin. Inf. Process.* **2016**, *30*, 172–177.
5. Wei, L.L.; Yang, W. The Study of Network Information Security Based on Information Filtering Technology. *Appl. Mech. Mater.* **2014**, *644–650*, 2978–2980. [CrossRef]
6. Kodialam, M.; Lakshman, T.V.; Sengupta, S. Configuring networks with content filtering nodes with applications to network security. In Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005; pp. 2395–2404.
7. Qiao, L.; Zhang, R.T.; Zhu, C.Y. personalized recommendation algorithm based on situation awareness. In Proceedings of the International Conference on Logistics, Informatics and Service Sciences, Barcelona, Spain, 27–29 July 2015; pp. 1–4.
8. Thorat, P.B.; Goudar, R.M.; Barve, S. Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *Int. J. Comput. Appl.* **2015**, *110*, 31–36.
9. Kohei, H.; Takanori, M.; Masashi, T.; Kawarabayashi, K.I. Real-time Top-R topic detection on twitter with topic hijack filtering. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 10–13 August 2015; pp. 417–426.
10. Lever, J.; Gakkhar, S.; Gottlieb, M.; Rashnavadi, T.; Lin, S.; Siu, C.; Smith, M.; Jones, M.R.; Krzywinski, M.; Jones, S.J.M.; et al. A collaborative filtering-based approach to biomedical knowledge discovery. *Bioinformatics* **2017**, *34*, 652–659. [CrossRef]
11. Shen, Y.F.; Shen, Y.W. Typed N-gram for Online SVM Based Chinese Spam Filtering. *J. Chin. Inf. Process.* **2015**, *29*, 126–132.
12. Huang, W.M.; Mo, Y. Chinese Spam message filtering based on text weighted KNN algorithm. *Comput. Eng.* **2017**, *43*, 193–199.
13. Chang, C.Y.; Lee, S.J.; Lai, C.C. Weighted word2vec based on the distance of words. In Proceedings of the 2017 International Conference on Machine Learning and Cybernetics, Ningbo, China, 9–12 July 2017; pp. 563–568.
14. Almas, Y.; Abdureyim, H.; Yang, C. Uyghur Text Filtering Based on Vector Space Model. *J. Xinjiang Univ. (Nat. Sci. Ed.)* **2015**, *32*, 221–226.

15. Zhao, X.D.; Aizezi, Y. A Uyghur bad text information filtering scheme based on mutual information and Cosine similarity. *Electron. Des. Eng.* **2016**, *24*, 109–112.
16. Dharmapurikar, S. Fast and scalable pattern matching for content filtering. In Proceedings of the ACM/IEEE Symposium on Architecture for Networking & Communications Systems, Princeton, NJ, USA, 26–28 October 2005; pp. 183–192.
17. Sherkat, E.; Farhoodi, M.; Yari, A. A new approach for multi-pattern string matching in large text corpora. In Proceedings of the International Symposium on Telecommunications, Tehran, Iran, 9–11 September 2014; pp. 72–77.
18. Hung, C.L.; Lin, C.Y.; Wu, P.C. An Efficient GPU-Based Multiple Pattern Matching Algorithm for Packet Filtering. *J. Signal Process. Syst.* **2017**, *86*, 1–12. [CrossRef]
19. Dawut, Y.; Abdureyim, H.; Yang, N.N. Research on Multiple Pattern Matching Algorithm for Uyghur. *Comput. Eng.* **2015**, *41*, 143–148.
20. 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.
21. Song, H.; Shi, N.S. Comment Object Extraction Based on Pattern Matching and Semi-supervised Learning. *Comput. Eng.* **2013**, *39*, 221–226.
22. Shao, K.; Yang, C.L.; Qian, L.B.; Fang, S. Structured Information Extraction Based on Pattern Matching. *Pattern Recognit. Artif. Intell.* **2014**, *27*, 758–768.
23. Sonal, G.; Christopher, D.M. Improved Pattern Learning for Bootstrapped Entity Extraction. In Proceedings of the Eighteenth Conference on Computational Language Learning, Baltimore, MD, USA, 26–27 June 2014; pp. 98–108.
24. Hojjat, E.; Hossein, S.; Ahmad, A.B.; Maryam, H. A Pattern-Matching Method for extracting Personal Information in Farsi Content. *U.P.B. Sci. Bull. Ser. C* **2016**, *78*, 125–138.
25. Cheng, C.; Yin, H.; Wang, L.S. A study of opinion question sentence classification in Question & Answering system. *Microcomput. Inf.* **2009**, *25*, 166–168.
26. Yu, Z.T.; Han, L.; Mao, C.L.; Deng, J.H.; Zhang, C. Answer extracting based on pattern learning and pattern matching in Chinese question answering system. *J. Comput. Inf. Syst.* **2007**, *3*, 957–964.
27. Tian, W.D.; Zu, Y.L. Answer extraction scheme based on answer pattern and semantic feature fusion. *Comput. Eng. Appl.* **2011**, *47*, 127–130.
28. Tohti, T.; Musajan, W.; Hamdulla, A. Uyghur Semantic String Extraction Based on Statistical Model and Shallow Linguistic Parsing. *J. Chin. Inf. Process.* **2017**, *31*, 70–79.
29. Achar, A.; Ibrahim, A.; Sastry, P.S. Pattern-growth based frequent serial episode discovery. *Data Knowl. Eng.* **2013**, *87*, 91–108. [CrossRef]
30. Muhammad, J.; Ibrahim, T.; Omar, H. Research of Uyghur Person Names Recognition Based on Statistics and Rules. *J. Xinjiang Univ. (Nat. Sci. Ed.)* **2014**, *31*, 319–324.
31. Yusuf, H.; Zhang, J.J.; Zong, C.Q.; Hamdulla, A. Name Recognition in the Uyghur Language Based on Fuzzy Matching and Syllable -character Conversion. *J. Tsinghua Univ. (Sci. Technol.)* **2017**, *57*, 188–196.
32. Zhang, L.; Wang, D.W.; He, L.T.; Wang, W. Improvement on Wu-manber multi-pattern matching algorithm. In Proceedings of the 3rd International Conference on Computer Science and Network Technology, Dalian, China, 12–13 October 2013; pp. 608–611.
33. Enwer, S.; Xiang, L.; Zong, C.Q.; Pattar, A.; Hamdulla, A. A Multi-strategy Approach to Uyghur Stemming. *J. Chin. Inf. Process.* **2015**, *29*, 204–210.
34. Abulimiti, M.; Aimudula, A. Morphological Analysis Based Algorithm for Uyghur Vowel Weakening Identification. *J. Chin. Inf. Process.* **2008**, *22*, 43–47.
35. Jiang, W.B.; Wang, Z.Y.; Yibulayin, T.; Liu, Q. Directed Graph Model of Uyghur Morphological Analysis. *J. Softw.* **2012**, *23*, 3115–3129.
36. Jiang, W.B.; Wang, Z.Y.; Yibulayin, T. Lemmatization of Uyghur Inflectional Words. *J. Chin. Inf. Process.* **2012**, *26*, 91–96.
37. Vasudha, B.; Vikram, G. Efficient Wu Manber String Matching Algorithm for Large Number of Patterns. *Int. J. Comput. Appl.* **2015**, *132*, 29–33.
38. Yan, J.J. Mechanism of ontology semantic extension with constraints for information filtering. *J. Comput. Appl.* **2011**, *31*, 1751–1755.

39. Li, X.; Xie, H.; Li, L.J. Research on Sentence Semantic Similarity Calculation Based on Word2vec. *Comput. Sci.* **2017**, *44*, 256–260.
40. Yibulayin, M.; Abulimiti, M.; Hamdulla, A. A Minimum Edit Distance Based Uighur Spelling Check. *J. Chin. Inf. Process.* **2008**, *22*, 110–114.
41. Maihefureti; Wumaier, A.; Aili, M.; Yibulayin, T.; Zhang, J. Spelling Check Method of Uyghur Languages Based on Dictionary and Statistics. *J. Chin. Inf. Process.* **2014**, *28*, 66–71.
42. Luo, Y.G.; Li, X.; Jiang, T.H.; Yang, Y.T.; Zhou, X.; Wang, L. Uyghur Lexicon Normalization Method Based on Word Vector. *Comput. Eng.* **2018**, *44*, 220–225.
43. Liu, Y.B.; Shao, Y.; Wang, Y.; Liu, Q.Y.; Guo, L. A Multiple String Matching Algorithm for Large-Scale URL Filtering. *Chin. J. Comput.* **2014**, *37*, 1159–1169.
44. Shen, F.X.; Zhu, Q.M. Text Information Filtering System Based on Adaptive Learning. *Comput. Appl. Softw.* **2010**, *27*, 9–10.
45. Li, Q.X.; Wei, H.P. Research of the Information Filtering Based on clustering Launched Classification. *Electron. Des. Eng.* **2014**, *22*, 14–16.
46. Li, Z.X.; Ma, Z.T. Research on Big Data Retrieve Filter Model for Batch Processing. *Comput. Sci.* **2015**, *42*, 183–190.
47. Tohti, T.; Hamdulla, A.; Musajan, W. Research on Web Text Representation and the Similarity Based on Improved VSM in Uyghur Web Information Retrieval. In Proceedings of the Chinese Conference on Pattern Recognition, Chongqing, China, 21–23 October 2010; pp. 984–988.
48. Tohti, T.; Musajan, W.; Hamdulla, A. Semantic String-Based Topic Similarity Measuring Approach for Uyghur Text Classification. *J. Chin. Inf. Process.* **2017**, *31*, 100–107.
49. Cheng, Z.; Zheng, D.; Li, S. Multi-pattern fusion based semi-supervised Name Entity Recognition. In Proceedings of the International Conference on Machine Learning & Cybernetics, Tianjin, China, 14–17 July 2013; pp. 45–50.
50. Xia, W.; Zhu, W.; Liao, B.; Chen, M.; Cai, L.J.; Huang, L. Novel architecture for long short-term memory used in question classification. *Neurocomputing* **2018**, *299*, 20–31. [CrossRef]
51. Rao, J.; He, H.; Lin, J. Experiments with Convolutional Neural Network Models for Answer Selection. In Proceedings of the International ACM Sigir Conference on Research & Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017; pp. 1217–1220.
52. Ravichandran, D.; Hovy, E. Learning surface text patterns for a question answering system. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, USA, 7–12 July 2002; pp. 41–47.
53. Kosseim, L.; Yousefi, J. Improving the performance of question answering with semantically equivalent answer patterns. *Data Knowl. Eng.* **2008**, *66*, 53–67. [CrossRef]
54. Zhang, C.X.; Gao, X.Y.; Lu, Z.M. Extract Reordering Templates for Statistical Machine Translation. *Int. J. Digit. Content Technol. Appl.* **2011**, *5*, 55–62.