



Article

Machine Translation Utilizing the Frequent-Item Set Concept

Hanan A. Hosni Mahmoud ¹  and Hanan Abdullah Mengash ^{2,*} 

¹ Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh P.O. Box 11671, Saudi Arabia; HAhosni@pnu.edu.sa

² Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh P.O. Box 11671, Saudi Arabia

* Correspondence: hamengash@pnu.edu.sa; Tel.: +966-11-823-8415

Abstract: In this paper, we introduce new concepts in the machine translation paradigm. We treat the corpus as a database of frequent word sets. A translation request triggers association rules joining phrases present in the source language, and phrases present in the target language. It has to be noted that a sequential scan of the corpus for such phrases will increase the response time in an unexpected manner. We introduce the pre-processing of the bilingual corpus through proposing a data structure called Corpus-Trie (CT) that renders a bilingual parallel corpus in a compact data structure representing frequent data items sets. We also present algorithms which utilize the CT to respond to translation requests and explore novel techniques in exhaustive experiments. Experiments were performed on specific language pairs, although the proposed method is not restricted to any specific language. Moreover, the proposed Corpus-Trie can be extended from bilingual corpora to accommodate multi-language corpora. Experiments indicated that the response time of a translation request is logarithmic to the count of unrepeated phrases in the original bilingual corpus (and thus, the Corpus-Trie size). In practical situations, 5–20% of the log of the number of the nodes have to be visited. The experimental results indicate that the BLEU score for the proposed CT system increases with the size of the number of phrases in the CT, for both English-Arabic and English-French translations. The proposed CT system was demonstrated to be better than both Omega-T and Apertium in quality of translation from a corpus size exceeding 1,600,000 phrases for English-Arabic translation, and 300,000 phrases for English-French translation.

Keywords: machine translation; frequent-item set; bilingual corpus; BLEU score



Citation: Mahmoud, H.A.H.; Mengash, H.A. Machine Translation Utilizing the Frequent-Item Set Concept. *Sensors* **2021**, *21*, 1493. <https://doi.org/10.3390/s21041493>

Academic Editor:
Ansar-Ul-Haque Yasar

Received: 30 November 2020
Accepted: 17 February 2021
Published: 21 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Machine Translation (MT) is an automated procedure of bilingual or multi-lingual translation [1]. There are several approaches to MT: linguistic (morphological), non-linguistic, and hybrid. Recently, statistical machine translation (SMT) and Neural Machine Translation (NMT) systems have been the leading machine translation paradigms [1–3]. Standard SMT techniques do not depend on any linguistic information, and do not apply any pre-processing procedures to generate the translation [4,5].

To face the challenges of machine translation, both pre-processing and post processing are utilized. Pre-processing can be accomplished via a number of lexical, syntactic, and semantic techniques. Lexical techniques include tokenization and normalization [6–8]. Syntactic pre-processing includes integrating additional linguistic information [9,10]. The used of named-entity notions is an example of semantic pre-processing. Pre-processing approaches are language-neutral and can be extended to other language pairs [9,10].

In this paper, we augment machine translation by using association mining concepts. Association mining can be utilized to develop efficient algorithms to analyze frequent sets representing a parallel linguistic corpus. As an analogy, it can be understood as a list of items, as in the original Apriori algorithm [11]. Co-occurring items are defined as item sets. The “support” of such an item set specifies how many groups are held in the item

set. If k market baskets contain $[X_1, X_2, X_3]$, the item set's support is k . The same analogy can be implemented where sentences are the "market baskets" and occurring ordered sets of words are the item sets. For instance, if M sentences contain [today is sunny], then the item set's support is M . Also, if M sentences contain [today is sunny], and their translation in language L is X , with support k , a rule can be created that "today is sunny" $\rightarrow X$ with support k .

In this paper, we introduce a new concept in translation statistical association rules. We treat words as item sets, and formulate rules based on frequent-set concepts. We formulate phrases in the corpus language as trie data structures. The whole corpus is built as a trie of the tries of phrases in the Corpus-Trie. From these tries, we can induce translation from language 1 \rightarrow language 2 depending on the frequency of occurrences. We devised an intelligent search technique that is linear in depth (D) of the CT. In this technique, there is no need to know anything in advance about the linguistic structure or grammatical rules of the languages. This methodology can be applied to any pair of languages; experiments here are in translation from English to Arabic, and from English to French.

One of the main contributions of our work is the building of the Corpus-Trie as a preprocessing step that is done offline. The translation process is transformed into a search process in the Corpus trie, which will be a fast procedure, has a time complexity in the order of \log the depth of the Corpus trie. Our concern is a preprocessing step to build a data structure based on association mining to enhance response time.

This paper presents a background and literature survey in Section 2. Section 3 presents the proposed system and a novel algorithm that can create a Corpus-Trie from a set of phrases. In Section 4, we propose the new concept of formulating a frequent-set trie representation and develop the notion of a Corpus-Trie. Section 5 introduces the experiments, demonstrating the computation time required to perform a typical translation request using the technique. It also investigates the costs of building a Corpus-Trie from a bilingual corpus. Conclusions are drawn in Section 6 with some ideas for future research.

2. Background and Literature Survey

We are concerned with the statistical machine translation paradigm, the association rule paradigm, and the research comparing them.

2.1. Statistical-Based Machine Translation (SMT)

SMT systems require huge text corpora to extract linguistic rules based on entropy [3,4,6]. SMT utilizes high-volume parallel corpora between source and destination languages, which are to a large extent available [7]. SMT proceeds from the assumption that every phrase in T (the target language) is a translation of a phrase in S (the source language) via a probabilistic formula. All pure SMT systems derive data from corpora that they have previously analyzed, and do not rely on linguistic information. SMT methods select the best representation. One of the crucial issues in SMT is the alignment problem. Alignment between phrases of the source and target languages has to be established. SMT relies on the use of statistics to solve the alignment problem and the induction of grammatical units [12–16].

2.2. Association Mining

Association rule formulation is a very effective technique which uses a data-mining paradigm to find patterns within large amounts of data [11,17–19]. This technique makes it feasible to find associations among different data items in large databases. We imported ideas from statistical machine translation to achieve a faster response. Association rules are generated from the dataset and reinforced by the support and confidence metrics [11]. In [17], the authors discuss the drawbacks of frequent item-set mining (FIM), such as high time complexity and high memory cost. They put forth an Array Prefix-Tree structure, which circumvents the need for recursion. They also present parallel data mining using systolic arrays. Work [18] presents parallel mining map-reduce. In [20], the authors utilize YAFIM, a parallel Apriori algorithm for memory-based mining. An incremental item-set

tree for data mining on data streams is introduced in [12]. Data mining techniques have rarely been combined with SMT. In [21], for example, the authors combine data mining techniques to distinguish cases of multiple parsing in machine translation of Indian languages. They introduce knowledge-based MT by utilizing text-mining techniques. They also introduced a sub-word augmented technique for derived sub-word level representations for text comprehension. Some authors devised parallel-sentence mining from a bilingual corpus, but not with a pure Statistical machine translation model. Source-target mapping of streaming data for MT was proposed using a variety of data mining [22–24].

2.3. Machine Translation and Data Mining

In [25], the author utilizes a machine learning approach to detect stack size, which is best for beam threshold runtime values for machine translation. In [26], the authors propose a sentiment analysis approach for MT. They analyze a machine-translated Bengali corpus to its original form and induce classifiers for translation. They focus on aspect-based sentiment analysis with emphasis on the Bangla language. In [27], the authors studied predictive data analytics and introduce a new concept, namely radius of neighbors, which was found to perform better than K-nearest neighbors in translation accuracy prediction. Work [28] introduces a knowledge-based machine translation system. It utilizes text mining by identifying semantic relations among entities present in text documents. In [29], the authors introduce a systematic approach for NMT and its application to context vectors. In [30], the authors present a graph-based approach for statistical translation.

2.4. Neural Machine Translation (NMT)

In [28], the authors present a framework that incorporates SMT word knowledge into NMT to address word-level obstacles. The NMT decoder performed accurate word prediction in both training and testing phases utilizing SMT. In [31], the authors use phrase-based SMT to calculate the cost of phrase-based decoding of NMT output and re-rank the n -best outputs. In [32], the authors survey parallel corpora and collect bilingual corpora; many corpora have 100,000 parallel sentences per language pair. Many papers discuss NMT, emphasizing zero shot neural machine [31–33] techniques. The authors of [34] note that NMT requires smaller data sizes, as small as a few thousand training sentences. In [35], an extensive survey for low resource NMT is introduced. Also in [36], the authors describe an analytical study and evaluation methods for multilingual machine translation as well as analytic evaluation matrices of machine translation.

2.5. Example-Based Machine Translation (EBMT)

The authors in [37] introduced the definition of the example used in EBMT. The main issues in EBMT are example acquisition, and base management. Also, it includes the notion of target sentence generation. EBMT adopted the concept of translation by analogy via example translations, which is the main core of the EMMT training process [38].

Example of a bilingual corpus:

English	Arabic
How much is that blue umbrella?	ما هو سعر الشمسية الزرقاء ؟
How much is that small bag?	ما هو سعر الحقية الصغيرة ؟

EBMT undergoes training using bilingual parallel corpora, which include sentence pairs. The example above shows a *minimal pair example*, where the sentences differ by only one element. These sentences make it simple to learn translations of sub-sentential units.

An EBMT will learn three aspects from those sentences in the bilingual corpora:

- “How much is that X?” Corresponds to “ما هو سعر س؟”
- “Red umbrella” corresponds to “الشمسية الزرقاء”
- “Small camera” corresponds to “الحقية الصغيرة”

The concepts in EBMT include training process and learning from example, which is different than our approach that does not include training or learning process. Our approach converts a bilingual corpus into a compact data structure namely the Corpus Trie, and converts the machine translation process into a search in association rules like mining.

2.6. Critique of Existing SMT Techniques

The response time for translation requests is crucial, especially for large requests; it is also a problem for real-time translation [39]. In spite of the presence of parallel corpora with alignment already annotated [40,41], searching this database to extract a phrase and the corresponding highest-probability translation associated with it requires scanning the corpus sequentially. The corpus may include millions of sentences. Response time can be reduced by reducing the corpus volume, but this will also reduce the accuracy of translation. Researchers have proposed several algorithms to expedite response time. For example, in [6], the authors introduce concept-formation techniques that group interrelated words, which can be helpful to reduce the complexity and time of association mining [42,43].

We propose a novel technique to represent the whole parallel corpus as a trie with frequencies attached to its edges. We store the corpus representation as tries which connect phrases from the source language and translated phrases from the target language, and store different translations and their frequencies. The space required for the trie is much less than the actual corpus (because repeated phrases will be stored once along with the highest-probability translated phrases from the target language), and the response time for any translation will be enhanced extensively, as the corpus will not need to be searched sequentially. Instead, the trie will be searched, and the time complexity will be in order with the trie's depth.

In this paper, we service the user's wish to utilize a bilingual corpus by submitting a translation request including the phrase Ph_S in the source language. We treat the corpus as a database of frequent phrase sets, and assume that the user will constrain the search to phrases that include the ordered words in Ph_S . A translation request may seek all frequent phrases containing word 1 and word 2, in order. In such cases, repeated search for all phrases would increase the response time in an unexpected manner. Therefore, in this research we emphasize pre-processing the corpus and inducing the trie representing it. We propose a data structure called Corpus-Trie and present novel algorithms that use CT to respond to translation requests, after exhaustive experiments.

3. Proposed Methodology

In this section, we introduce the proposed concept in translation statistical utilizing association rules. We treat words as item sets, and formulate rules based on frequent-set concepts. We formulate phrases in the corpus language as trie data structures. The whole corpus is built as a trie of the tries of phrases in the Corpus-Trie. From these tries, we can induce translation from language 1 \rightarrow language 2 depending on the frequency of occurrences.

We present the proposed CT system in Figure 1 clarifying the building blocks of the system. We start by building the Corpus Trie by reading new phrases with their translations and insert them in the CT if they do not exist before. Each phrase in the source language is represented as a trie and inserted into the CT in the appropriate position (if a subset of this phrase already exists in the CT). Each phrase in the source language will be associated with multiple translations from the target language; these multiple translation will be stored in a tree-structure namely the Z-tree. The CT building process should be done offline since it is a lengthy process, as indicated in the experimental results in Section 5.

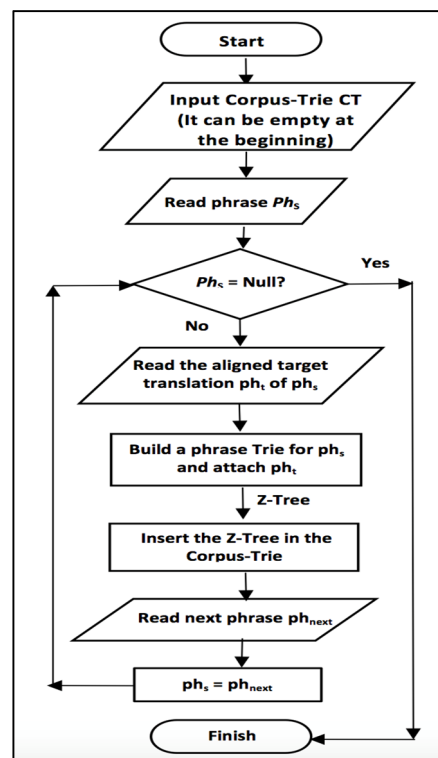


Figure 1. Flowchart of the CT system.

3.1. Building a Phrase Trie

We also introduce several algorithms to build the phrase trie-like structure as shown in Algorithm 1 in Section 3.1. Z-tree insertion is introduced in Algorithm 2, which describes the Z-tree insertion of a new translation in the Z-tree. The Z-tree is a two-level tree and is defined as having a root node and leaf nodes. Leaves are defined as an ordered pair: <content, frequency>. All the translations and their frequencies of content (parent) are added as a tree.

Algorithm 1. Build a phrase trie (Input: phrase Ph , $length(Ph) n$; Output: Trie T_P).

Start

// phrase Ph consists of an ordered tuple of n words: $w_i \forall i = 1$ to n

// T_P is empty

Insert w_1 as the root node of the tree T_P

Parent = root;

For $i = 1$ to n

$w_{i+1} = w_i + w_i + 1$;

Add a left child node: N_{left} to parent;

Content (N_{left}) = $w_i + 1$;

Parent = N_{left} ;

Get all translation of the content (parent) from the target language T

/ Translation from the Target language T will be obtained using statistical translation machine or neural machine translation machine */*

Save them to the array S_Z

For $j = 1$ to size of (S_Z)

Call the procedure Z-Tree-insertion ($j, S_Z[j]$)

End for

End for

End

Algorithm 2. Z-Tree-insertion (input $j, S_Z [j]$).

```

Start
 $i = 1;$ 
Step 1:
If node  $[j] = \text{empty}$ 
Then
{Insert  $S_Z [j]$  at node  $[j]$ ;
 $\text{Frequency}(j) = \text{Frequency}(j) + 1;$ }
Else if ( $S_Z [j] = \text{content}(\text{node}[j])$ )
Then
 $\text{Frequency}(i) = \text{Frequency}(i) + 1;$ 
Else if ( $S_Z [j] < \text{content}(\text{node}[j])$ )
Then
 $\{j = j + 1;$ 
Go to step 1;}
Else if ( $S_Z [j] > \text{content}(\text{node}[j])$ )
Then
{Insert a new empty node, node  $[x]$  between nodes: node  $[j]$  and node  $[j + 1]$ ;
 $\text{Content}(\text{node}[x]) = S_Z [j];$ 
 $\text{Frequency}(\text{node}[x]) = 1;$ 
 $j = j + 1;$ 
Go to step 1}
End if
End

```

3.2. Building the Corpus-Trie

We will use Algorithm 3 to build the Corpus-Trie. In order to facilitate the algorithm, we will define a phrase as a prefix of another prefix in Definition 1.

Definition 1. Ph_{S1} is a prefix of Ph_{S2} : The symbols Ph_{S1} and Ph_{S2} denote phrases, and both of them are an ordered tuple. Ph_{S1} is a prefix of Ph_{S2} iff $Ph_{S1} = [w1, w2, \dots, wm]$, $Ph_{S2} = [w1, w2, \dots, wn]$, where $m \leq n$. The algorithm that builds the Corpus-Trie is an incremental algorithm built through a sequence of phrase insertions. The Corpus-Trie itself is built through a single pass of the database.

Algorithm 3. Build Corpus-Trie CT (input $Ph_S = \{Ph_{S1}, Ph_{S2}, \dots, Ph_{Si}, Ph_{Si+1}, \dots, Ph_{Sn}\}$).

```

/* Consider the parallel corpus. In the source language the corpus consists of sub phrase as follows:
 $Ph_S = \{Ph_{S1}, Ph_{S2}, \dots, Ph_{Si}, Ph_{Si+1}, \dots, Ph_{Sn}\}$  */
Start
Have a root node  $n0 = \hat{R}$  which is an empty phrase
Read  $Ph_{S1}$ 
 $n1 = n0.\text{child}$ 
 $\text{Content}(n1) = Ph_{S1}$ 
For  $i = 1$  to  $n$ 
Read  $Ph_{Si+1}$ 
If  $Ph_{Si}$  is a prefix of  $Ph_{Si+1}$ 
Then
{Append  $n_{i+1}$  to be a node child to  $n_i$ 
 $\text{Content}(n_{i+1}) = Ph_{Si+1}$ }
Else
{Append  $n_{i+1}$  to be a node child to  $n_{i-1}$ 
 $\text{Content}(n_{i+1}) = Ph_{Si+1}$ }
End if
End for
End

```

Adding a phrase to the Corpus-Trie is clarified in Example 1.

Example 1. Figure 2 shows the trace of Algorithm 1 processing the following phrases from a bilingual corpus: $Ph = \{[he], [has, money], [he, has], [he, has, to], [he, has, put], [he, has, to, put, money], [he, has, to]\}$.

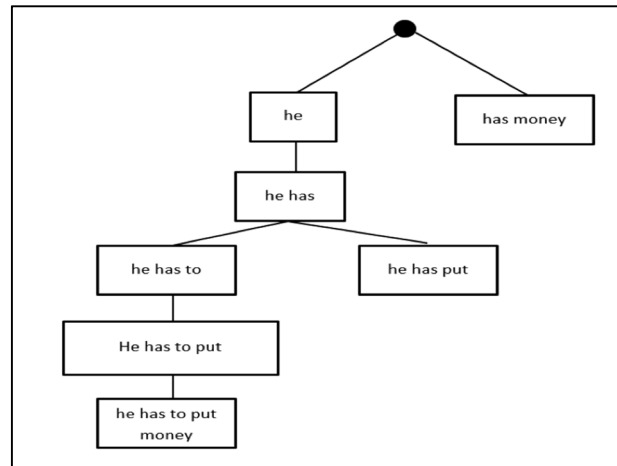


Figure 2. The CT illustration of Example 1.

The following steps are performed:

1. An empty node is defined as the root node (node 0);
2. The first phrase [he] is defined as an ancestor to the root node (left child);
3. The second phrases, [has, money] and [he], are not identical. They have an empty subset. Therefore, a new child node is created and added as the right node because “has” is alphabetically greater than “he”;
4. The third phrases, [he, put] and [has, money], are not identical, as they have an empty subset. So recursively we will compare the third phrase, [he, put] and [he] are not identical, but have [he] as a subset and [he] is a prefix of [he, put]. Therefore, a left child node is created to the parent node [he];
5. The fourth phrase, [he, has, to] has a common subset with [he, has], and [he, has] is a prefix of [he, has, to], so it will be added as a left child node to [he, has]. This is added to the left child because [he, has] is a leaf node (node 4);
6. The fifth phrase, [he, has, put] has a common subset with [he, has], and [he, has] is a prefix of [he, has, put], so it will be added as a left child node to [he, has]; left child, because [he, has] is a leaf node;
7. The sixth phrase, [he, has, to] has a common subset with [he, has], and [he, has] is a prefix of [he, has, to], so it will be added as a right child node to [he, has];
8. The seventh phrase, [he, has, to, put, money] has a common subset with [he, has, to], and [he, has, to] is a prefix of [he, has, to, put, money], so it will be added as a left child node to [he, has, to]; left child because [he, has, to] is a leaf node;
9. The eighth phrase, [he, has, to] is identical to node 4 so it will not be added.

Lemma 1. (Corpus-Trie size): Let CT denote the Corpus-Trie built from a bilingual corpus of N distinct phrases of an average of m words per phrase, using Algorithm 1:

1. The upper bound of the nodes' count in the CT is $m \times N + 1$;
2. The upper bound of the number of layers in the CT is $m \times N + 1$ (worst case).

Proof (from definition). The memory required to store the Corpus-Trie is much less than the size of the bilingual corpus, because it stores the repeated phrases just once, and stores only the phrase translations with the highest frequencies. Therefore, the methodology does not depict an overhead in real-world domains. \square

In Figure 3a, we show a flowchart of the translation process, which mimics a search problem in the Corpus Trie, Algorithm 4 provides more details of the translation problem.

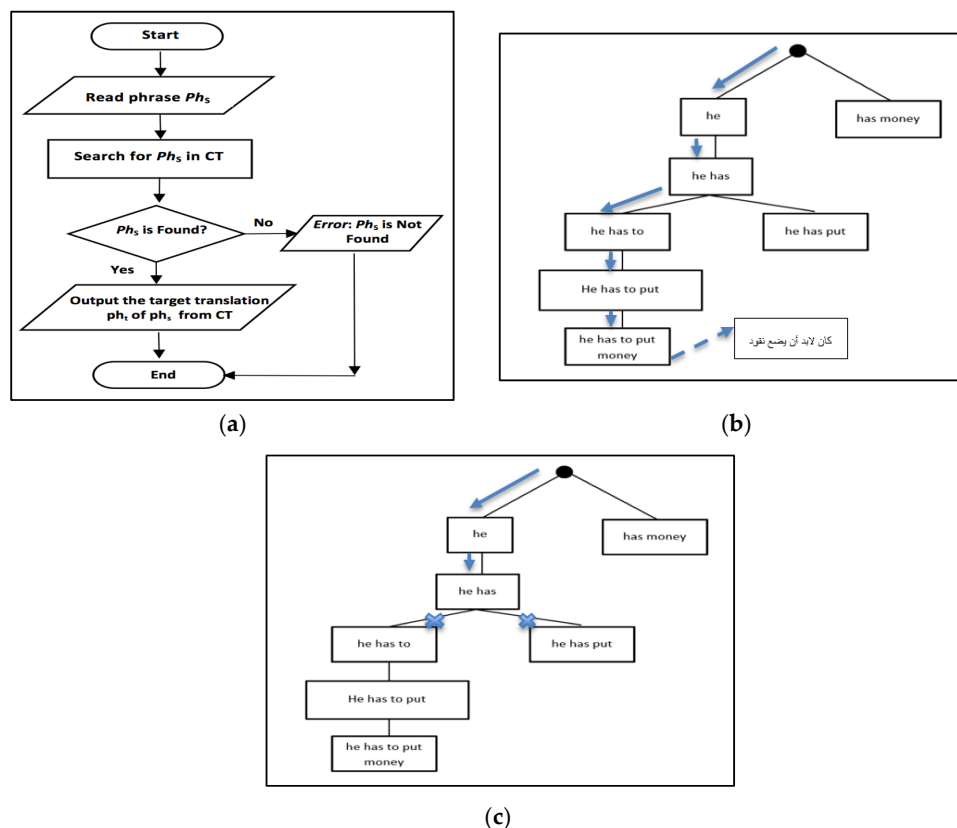


Figure 3. (a) Flowchart of the translation process; (b) An example showing the translation of the sentence “he has to put money”; (c) An example showing the translation of non-existent sentence in the Corpus Trie “he has money”.

Algorithm 4 is the search algorithm, the translation process using the Corpus-Trie. It starts with the input phrase S from the source language, and divides it into words $w_1, w_2, w_3, \dots, w_m$. The word w_i ($i = 1$ to m) is searched in the trie starting from the root of the Corpus-Trie until it is located, then the next word in S is searched, and so on continuing down the trie, if found.

Figure 3b, presents an example showing the translation of the sentence “he has to put money”, from the Corpus Trie shown previously in Figure 2. Another example of trying to translate a phrase that is not found in the Corpus Trie is shown in Figure 3c.

Algorithm 4. Get Translation (input key: Ph_s , output node: Node).

```

Start
For  $I = 1$  to  $N$  //  $N$  is number of words  $w(i)$  in key:
  If  $w(i)$  in  $node.children$ :
     $node = node.children[w(i)]$ 
  Else:
    Return None
  End if
End For
Return  $node.Ph_T$ 
End
  
```


4. The Notion of a Corpus-Trie

In this section, we propose a formulation for parallel corpora, without any emphasis on particular languages. The defining trait of a bilingual corpus is translation between two languages: the source language S and the target language T . Word alignment should be part of the corpus. Word alignment is done on biphrases, by our algorithm, using a bilingual dictionary to align each word in the sentence in the source language to its match in the target language.

Later we will extend the formulation to “source language \Rightarrow multi-target language” translation. To formulate our technique, we define a translated phrase formally and recursively as a prefix-trie structure. We also introduce formulated definitions for a corpus and a Corpus-Trie in Definitions 2, 3, and 4. In addition, we describe a data structure called a Z-tree in Definition 5. Also, in Lemma 2, we prove that a phrase is a prefix trie-like structure. In Figure 4a, the trie-like structure of a phrase Ph is shown for a phrase Ph of four words $\langle w_1, w_2, w_3, w_4 \rangle$. In Definition 5, we define an extended version of phrase Ph to include Ph_S (phrase of the source language) and the corresponding Ph_T and $AL_{S:T}$ (phrase in the target language) as depicted in Figure 4b.

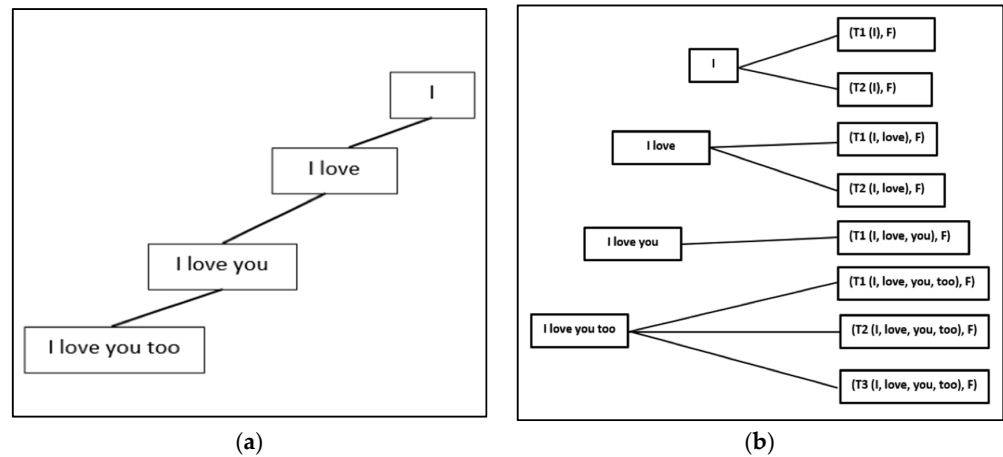


Figure 4. (a) Trie-Like structure of a phrase Ph is shown for a Phrase Ph of four words $\langle I, \text{love}, \text{you}, \text{too} \rangle$; (b) Phrase Ph that includes Ph_S (phrase of the source language) and the corresponding Ph_T (phrase in the target language).

Definition 2. A translated phrase Ph :

Define Ph as an ordered tuple $Ph = \langle Ph_S, \langle Ph_T, AL_{S:T} \rangle \rangle$, where:

- Ph_S and Ph_T are two phrases from source language and target language, respectively; Ph_T is the corresponding phrase translation from the target language, with alignment $AL_{S:T}$ between phrases Ph_S and Ph_T ;
- M is the set of all words constituting Ph_S . All words in Ph_S have an order-by-word location in ascending order. For any two words, the i th and j th words are ordered as $i < j$, i.e., $w_i < w_j$, where w_i represents the word in position i ;
- Ph_S is itself defined as Ph , which yields a recursive definition of Ph ;
- Ph can be represented as a trie, Ph -trie (see proof in Lemma 2).

Definition 3. The relation between Ph , Ph_S , Ph_T and AL

- Ph_S is a recursive definition of other phrases as follows: $Ph_S = \langle Ph_{S1}, \langle Ph_{T1}, AL_{S1:T1} \rangle \rangle$;
- Therefore Ph can be defined as follows: $Ph = \langle Ph_S, \langle \langle Ph_{S1}, \langle Ph_{T1}, AL_{S1:T1} \rangle \rangle, AL_{S:CT} \rangle \rangle$, where Ph_S is a prefix of Ph_{S1} , which means if Ph_S has n words, then Ph_{S1} has at least $n + 1$ words.
- Ph can be defined further as: $Ph = \langle Ph_S, \langle \langle Ph_{S1}, \langle \langle Ph_{S2}, \langle Ph_{T2}, AL_{S1:T1} \rangle \rangle, AL_{S1:T1} \rangle \rangle, AL_{S:CT} \rangle \rangle$, and so on.

Definition 4. A Corpus-Trie:

1. A parallel corpus C is a set of translated Phrases Ph ;
2. A Corpus-Trie CT is a trie representation of C . therefore CT is a trie;
3. A Corpus-Trie CT is built using Definition 3 recursively from phrases;
4. Since each phrase Ph (from Lemma 2) is a trie, the Corpus-Trie CT is a Trie;

Definition 5. The Z-tree:

A Z-tree is a tree in the third dimension; its root is a node in the Ph-trie of the phrase Ph_S in the source language, and it has only one level, in which leaf nodes are several translated phrases Ph_T from the target language with their frequencies in the target corpus.

Lemma 2. A phrase Ph is a prefix, trie-like, and only branches one-sided (left-sided); the whole phrase Ph is at the only leaf node of the trie.

Proof of Part 1. Since Ph can be defined as follows:

$Ph = \langle Ph_S, \langle \langle Ph_{S1}, \langle \langle Ph_{S2}, \langle Ph_{T2}, AL_{S1:T1} \rangle \rangle, AL_{S1:T1} \rangle \rangle, Ph_T, AL_{S:CT} \rangle \rangle$, disregard at this point the translation of Ph (the phrase Ph_{TS} and its word alignment).

1. The trivial case: Let Ph have a single word w , i.e., $Ph = \langle Ph_S \rangle$, and $Ph_S = w$. Then Ph can be defined as a trie with one node n representing w ;
2. Let Ph contain two words $w1$ and $w2$. Then, it is defined as a trie of two nodes $n1$ and $n2$, where node $n1$ contains $w1$ and node $n2$ contains $w1$ and $w2$ (for a true trie, node $n2$ will contain word $w2$ only);
3. For the last case, Ph , which contains m words, can be represented by a trie-like representation of m nodes; the root will contain $w1$, the second node will contain $w1, w2$, and so on, until the m -th node, which will contain words $w1, w2, \dots, wm$. \square

Proof: of Part 2. In the first part, we proved that the last node m would contain all the words in Ph as ordered. Therefore, the last node will contain Ph . \square

Definition 6. Phrase Ph , including Ph_S and the corresponding $Ph_T, AL_{S:CT}$

$Ph = \langle Ph_S, \langle \langle Ph_{S1}, \langle \langle Ph_{S2}, \langle Ph_{T2}, AL_{S1:T1} \rangle \rangle, AL_{S1:T1} \rangle \rangle, Ph_T, AL_{S:CT} \rangle \rangle$

1. The trivial case: Let Ph contain only one word and its translation. Assume that the alignment is null; there is no alignment because it is a single word in the source language and one translated word in the target language, i.e., $Ph = \langle Ph_S, Ph_T \rangle$. Ph has a one-to-one relation to Ph_S , but Ph has a one-to-many relationship to Ph_T , because the same word can be translated to one or more words in the target language. $Ph \rightarrow Ph_S$ where the arrow notation \rightarrow is used to define one-to-one relationships, and the double arrow $\rightarrow\rightarrow$ is used to define one-to-many relations. The definition will be summarized using the arrow notation as follows: $Ph \rightarrow Ph_S \rightarrow\rightarrow Ph_T \rightarrow\rightarrow AL_{S:CT}$; Then, Ph can be represented as a trie with one node n representing $Ph_S = w$, and a tree in the the Z-dimension, with nodes that include the sets of ordered pair $S_Z = (\langle Ph_T, AL_{S:CT} \rangle)$. One node, representing one of the elements of the set S_Z , is labeled with the frequency of this Ph_T in the target corpus T and translates Ph_S ;
2. Let Ph contain two words $w1$ and $w2$; then it can be represented by a trie (Ph-trie) of two nodes $n1$ and $n2$. Node $n1$ contains $w1$ and node $n2$ contains the ordered tuple $\langle w1, w2 \rangle$. In the Z-dimension, each node in Ph-trie will be a root node to a Z-tree including the different translations of the content of node X , in the target language with their frequencies, a set of ordered tuple $S_Z = \langle X_T, AL_{S:CT}, F \rangle$ where F represents the frequency of X_T in T and translates to X ;
3. The general case Ph which contains m words can be represented by a trie-like structure of m nodes. The first node contains $w1$, the second node contains $w1, w2$ and so on until the m -th node which contains words $w1, w2, \dots, wm$. Moreover, a Z-tree, in the third dimension, is built for all nodes.

Figure 4a,b depicts an example clarifying Definition 6, for a phrase Ph of four words $\langle I, love, you, too \rangle$. T_i represents a translation of word w_i with frequency F .

5. Experimental Results

Experiments emphasized the cost of computation of the proposed method and whether translation request answering is adequately fast. The goal was to establish if the construction of the Corpus-Trie is affordable (though it is done offline). It was demonstrated that the translation request processing is fast and that Corpus-Tries have $O(\log N)$. N is noted as the count of phrases of the source language (excluding repetition); this is because the values of the nodes in the trie are sorted in each horizontal level.

5.1. Experimental Data

The United Nations Parallel Corpus v1.0 is composed of official records and other parliamentary documents of the United Nations, which are in the public domain. These documents are generally available in the six official languages of the United Nations. The corpus includes sentence-level alignments and allows access to multilingual corpora in various natural languages. We used the English-Arabic parallel corpus presented in [44]. It contains 456,552,223 pairwise-aligned English-Arabic sentences. We used two million of those pairs for our experiments. Building the general Corpus-Trie was done offline. Algorithm 1 was used to create Corpus-Tries from the sentences chosen from the bilingual corpus. We built nine different Corpus-Tries using 200,000 sentence pairs, with an increment of 200,000 sentence pairs for the next trie.

We also used the English-French parallel corpus presented in [44]. We used 500,000 pairwise-aligned English-French sentence pairs for our second set of experiments. Building the general Corpus-Trie was done offline. Algorithm 1 was used to create Corpus-Tries from the sentences chosen from the bilingual corpus. We built five different Corpus-Tries using 100,000 sentence pairs, with an increment of 100,000 sentence pairs for the next trie.

5.2. Building a Corpus-Trie

Having proven that Corpus-Trie aids fast translation-request processing, we wanted to establish that the time required to construct a CT from a database-phrase bilingual corpus is reasonable (see Figure 5a).

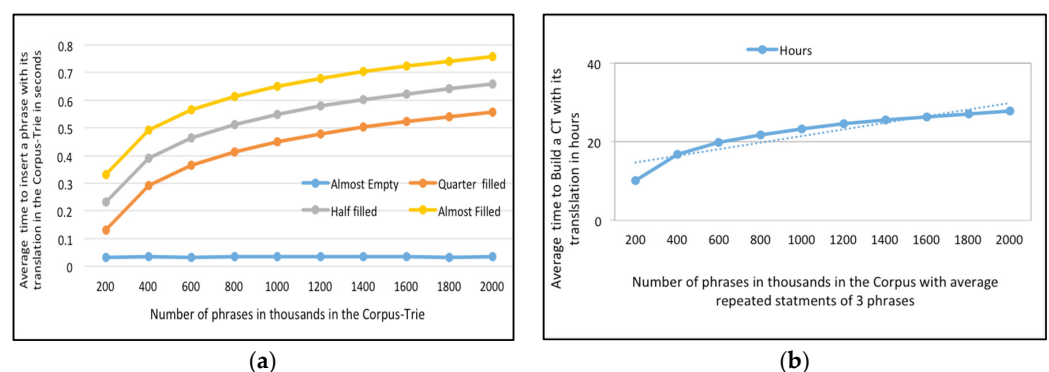


Figure 5. (a) Average time to insert a phrase with its translation in the Corpus-Trie in seconds versus Number of phrases in thousands in the Corpus-Trie for the English-Arabic parallel corpora; (b) The cost of *Corpus-Trie* construction (the number of phrases which are repeated in the corpus is in the average of three phrases), for the English-Arabic parallel corpora.

Lemma 1 established that the cost of inserting N phrases has an upper bound of $O(N \log n)$. We measured the CPU time that Algorithm 1 required to convert a corpus-phrase bilingual corpus into a Corpus-Trie. Inserting a phrase into the Corpus-Trie required two steps, the first one being to search for the phrase, and the second step being dependent on the first step: either the phrase was not found, or the whole phrase or a portion of it

was found. If the whole phrase was found, it did not have to be inserted, otherwise one or more nodes had to be created to insert it (see Figure 5b).

In Figure 6a, we measured the CPU time that Algorithm 1 requested to convert a corpus-phrase English-French corpus into a Corpus-Trie. In Figure 6b, we present the cost of Corpus-Trie construction (the number of phrases which are repeated in the corpus is an average of three) for the English-French parallel corpus.

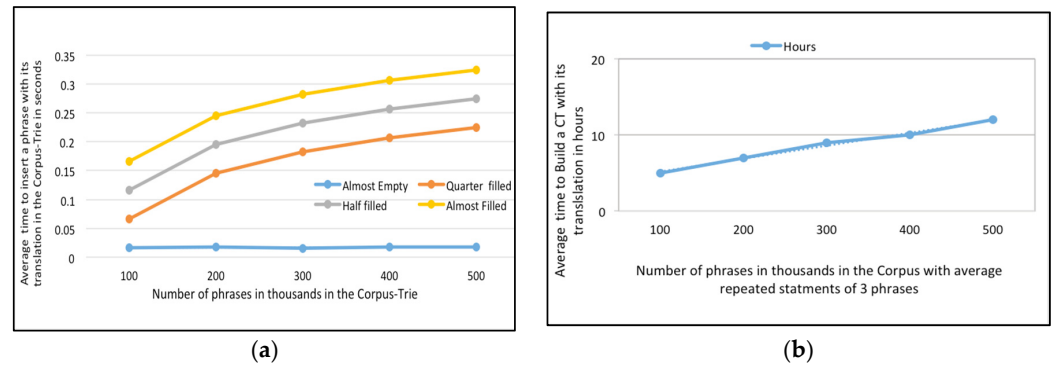


Figure 6. (a) Average time to insert a phrase with its translation in the *Corpus-Trie* in seconds versus Number of phrases in thousands in the *Corpus-Trie* for the English-French parallel corpora; (b) The cost of *Corpus-Trie* construction (the number of phrases which are repeated in the corpus is in the average of three phrases) for the English-French parallel corpora.

5.3. Translation Request

In this subsection we introduce two types of experiments: the first type is the computation of the average cost of answering a translation request using different sizes of test data and corpus tries. The second type is calculating the average error rate of the translation process. The test data and the experiments are described in the following subsections.

5.3.1. Test Data

We carried two types of experiments, for type I: our test data for both English-Arabic and English-French translations consists of six sets of 6000 English sentences each. Each set contains sentences of length equal to five words, seven words, 10 words, 13 words, 16 words, and 18 words respectively. Each set contains 100% of the sentences from the Corpus Trie. For experiment of type II: each set contains 90% of the sentences from the Corpus Trie, and 10% of the sentences that do not exist in the Corpus Trie, but either as a whole sentence or as an ordered subset of an existing sentence.

5.3.2. Experiment Type I: Cost of Answering a Translation Request

In this type of experiments, the computational costs of answering a translation request were computed as an average by the node count in the Corpus-Trie that Algorithm 3 has to visit.

The first set of experiments utilizes 1000 random translation requests for each set of the test data (All of them are presented in the portion of the English-Arabic corpus); The average number of nodes visited per translation request for each of ten Corpus tries, of different sizes, are computed as shown in Figure 7.

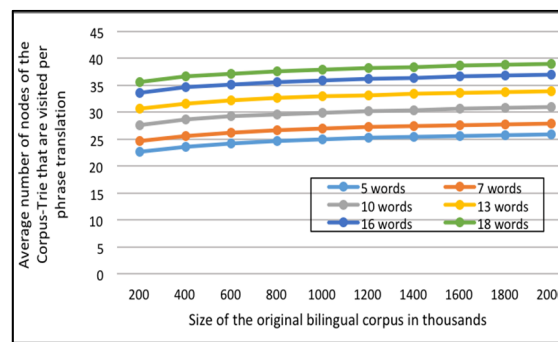


Figure 7. Average number of nodes of the *Corpus-Trie* that are visited per phrase translation versus Size of the original bilingual corpus in thousands, for the English-Arabic corpora.

The second set of experiments utilizes 1000 random translation requests for each set of the test data (All of them are presented in the portion of the English-French corpus); The average number of nodes visited per translation request for each of ten *Corpus* tries, of different sizes, are computed as shown in Figure 8.

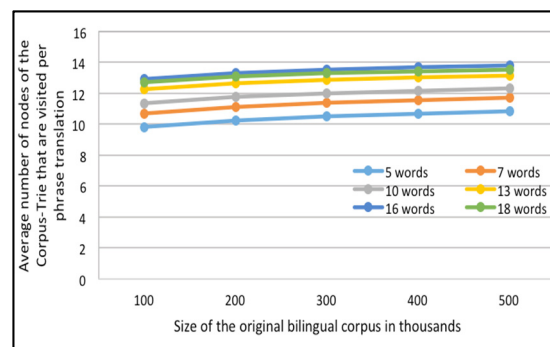


Figure 8. Average number of nodes of the *Corpus-Trie* that are visited per phrase translation versus Size of the original bilingual corpus in thousands, for the English-French corpora.

Figures 7 and 8 contain multiple curves: one each for requests of phrases of five words and up, to requests containing phrases of 18 words, for the English-Arabic corpus and the English-French corpus, respectively.

It can be established that when answering a translation request, the system will navigate only a small part of the *Corpus-Trie*. The count of visited nodes is less than the log of the number of distinct original bilingual corpora.

5.3.3. Experiment Type II: The Error Rate of The Translation Process

In this experiment, the error rate of the translation process is investigated for *Corpus Trie* of different sizes. We used 1000 phrases from each set that are randomly chosen from the data set, and we repeated the same experiment where 5000 phrases are randomly chosen for each set.

For the English-Arabic corpus, Figures 9 and 10 represent the error rate (i.e., the percentages of unfound phrases in the *Corpus-Trie*) per one thousand and five thousand phrases from each set respectively.

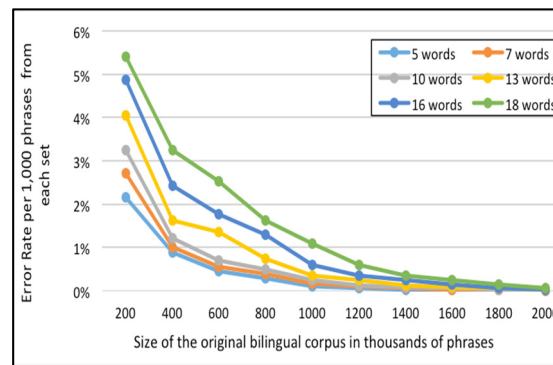


Figure 9. Error rate per 1000 phrases from each set (un-found phrases in the *Corpus-Trie*), for the English-Arabic corpora.

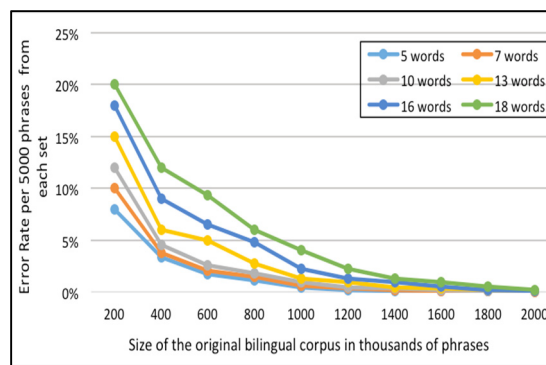


Figure 10. Error rate per 5000 phrases from each set, for the English-Arabic corpora.

As indicated in Figures 9 and 10, the percentage of failure to locate the phrase in the *Corpus-Trie* decreases with the increase of corpus size, and approaches zero with a corpus size of two million phrases.

For the English-French corpus, Figures 11 and 12 represent the error rate (i.e., the percentages of unfound phrases in the *Corpus-Trie*) per one thousand and five thousand phrases from each set of the test data respectively.

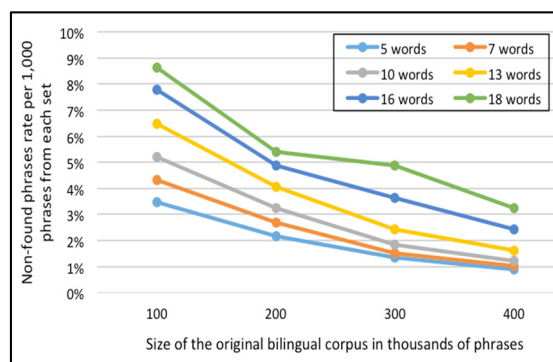


Figure 11. Error rate per 1000 phrases of each set, for the English-French corpora.

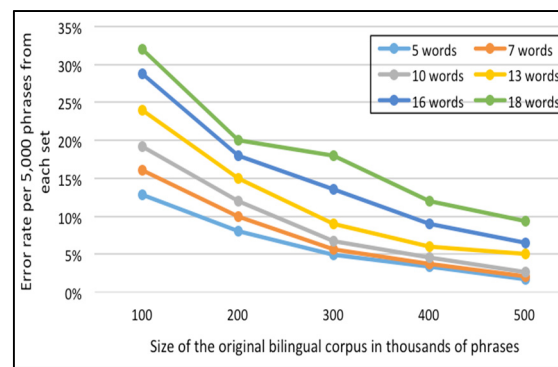


Figure 12. Error rate per 5000 phrases of each set, for the English-French corpora.

As indicated in Figures 11 and 12, the percentage of failure to find the phrase in the Corpus-Trie decreases with the increase of corpus size, and approaches zero with a corpus size of 500 thousand phrases.

5.4. Translation Quality Evaluation

To assess the translation quality of our proposed system, we utilized manual and automated translation quality metrics. We compared our system against two open source machine translation platforms. The first one is Omega-T, which is an open source platform that utilizes different translation approaches [45]. In our comparison we used the property of translation memories (TMEM) reuse, which basically is the reusing of previous translations. Reference translations can also be included in TMEM from manual translations as well as from other machine translation systems. Also, same subject TMEM can be utilized such as translating legal document; previous translated legal documents can be reused. For our comparison we imported part of the data set from [44] as a TMEM in Omega-T platform.

The second platform is Apertium, which is an open source software for machine translation (MT) that is rule-based [46]. It is used to construct MT systems for a diversity of languages. Apertium utilizes linguistic facts gathered from different languages. It also utilizes multilingual dictionaries and grammatical rules of semantic and syntactic nature.

A qualitative evaluation of machine translation output is done both manually and automatically. Manual evaluation is done mainly by comparing translations from human experts to the output of machine translation, using human judges. The manual evaluation metrics of comparing our proposed system versus Omega-T and Apertium by human translators, are: fluency, adequacy, meaning, and preference.

We also include two other measures namely understandability and fidelity. Fidelity is a measure of the information retention in the translation text compared to the original one. While fidelity is measured with reference to both the original text and the translated text separately, understandability is measured with reference to the translated text only.

The human translation expert will first examine the translated sentence. The source sentence is then presented and judges would rate the original sentence on how more information they gained from it. The amount of information they gained from the original sentence is inversely proportion to the translation quality.

Automated quality evaluation of machine translation is performed using both BLEU and METEOR systems. BLEU is a very well-known MT quality evaluation and it estimates precision. METEOR is also well known but more complicated measure which estimates both precision and recall using F_{mean} score [47,48]. In the following subsections, we discussed the quality evaluation of our proposed system versus Omega-T and Apertium.

5.4.1. Manual Evaluation of the Translation Quality

The translations from our proposed CT translation system were scored both manually and automatically. Three bilingual, native Arabic-speaking persons with master's degrees

or higher were asked to be volunteer evaluators. Each evaluator received an explanation of the scores. They made blind evaluations of three translation systems: System 1, System 2, and System 3, interchanged for each phrase translation. Omega-T [45], professional manual translation, and the CT system were compared. Each evaluator was asked to evaluate the same 100 phrase translations (20% were 7-word phrases, 20% were 10-word, 20% were 13-word, 20% were 16-word, and 20% were 18-word).

The evaluators were asked to evaluate phrases on Likert scales. They were asked to score four metrics: fluency, adequacy, meaning, and preference. Fluency was defined as an evaluation of readability ranging from 5 (perfect, “like reading an article”) to 1 (not understandable). Adequacy scores reflected evaluation of information conservation, ranging from 5 (100% information conservation) to 1 (0% information conservation). Meaning was defined as intent preservation, ranging from 5 (same meaning as the source phrase) to 1 (completely different meaning). The last measure was preference; an option was given to choose which translation was preferred using a two-answer scale of either 5 (strongly prefer) or 1 (do not prefer). Evaluators could give preference to one or more systems for each phrase translation. The results are presented in Table 1. The same experiment was carried out for the English to French translations; results are presented in Table 2.

Table 1. Manual evaluation of the translation quality, for the English-Arabic corpora.

Score	Translation System		
	Omega-T	Professional Translation	The Proposed CT Translation System
Fluency	4.35	4.8	4.5
Adequacy	4.5	5	4.6
Meaning	4.3	5	4.4
Preference	4.1	5	4.5
Understandability	4.2	5	4.7
Fidelity	3	0	0.8

Table 2. Manual evaluation of the translation quality, for the English-French corpora.

Score	Translation System		
	Omega-T	Professional Translation	The Proposed CT Translation System
Fluency	4.01	4.8	4.4
Adequacy	4.2	5	4.45
Meaning	4.2	5	4.36
Preference	4.0	5	4.35
Understandability	3.8	5	4.5
Fidelity	3.4	0.3	0.7

5.4.2. Automated Evaluation

Translation quality was also evaluated by an automatic process. Both BLEU and F_{mean} scores [47] were utilized. The BLEU score measures the precision of unigrams, up to four-grams, with respect to reference translations. BLEU measures accuracy, and takes values from zero to 100%; usually, a BLEU score of less than 15% implies bad translation, and a score of 50% is considered an excellent translation. The experiments were designed by comparing the average BLEU score of the proposed system against translations from Omega-T [45] and Apertium [46] translators. The results are shown in Tables 3 and 4 for English-Arabic translation and English-French translation, respectively.

Table 3. Automated evaluation using BLEU Score, for the English-Arabic corpora.

Translation System	Average BLEU Score		
	1000 Phrases	5000 Phrases	10,000 Phrases
Omega-T	41	42	45
Apertium Translator	41	43	44
CT translator (400,000 phrases)	39	40	39.5
CT translator (800,000 phrases)	42	43	43.5
CT translator (1,200,000 phrases)	44	45	46.3
CT translator (1,600,000 phrases)	46	46.7	46.5
CT translator (2,000,000 phrases)	49.2	49.9	51

Table 4. Automated evaluation using BLEU Score, for the English-French corpora.

Translation System	Average BLEU Score		
	1000 Phrases	5000 Phrases	10,000 Phrases
Omega-T	38	38.7	42
Apertium Translator	39	39.9	42.3
CT translator (100,000 phrases)	38.5	39	39.8
CT translator (300,000 phrases)	39	39.5	40.5
CT translator (500,000 phrases)	41	42	42.3

The results indicate that the BLEU score for the proposed CT system increases with the size of the number of phrases in the CT, for both English-Arabic and English-French translations. The proposed CT system was demonstrated to be better than both Omega-T and Apertium in quality of translation from a corpus size exceeding 1,600,000 phrases for English-Arabic translation, and 300,000 phrases for English-French translation.

Unlike BLEU, which only estimates precision, METEOR estimates precision and recall, and combines both using F_{mean} score [47,48]. Tables 5 and 6 present automated evaluations using the F_{mean} score for the English-Arabic corpus and the English-French corpus respectively. Experiments were designed to compare the average F_{mean} score of the proposed system with translations from Omega-T and Apertium Translator. The results indicated that the F_{mean} metric for the proposed CT system increases with the size of the number of phrases in the CT for both English-Arabic and English-French corpora. For English-Arabic translation, the proposed CT system was shown to be superior to both Omega-T and Apertium in quality of translation from all corpus sizes beginning with 400,000 phrases, and to be dramatically enhanced by increasing the corpus size to two million phrases.

Table 5. Automated evaluation using F_{mean} Score, for the English-Arabic corpora.

Translation System	Average BLEU Score		
	1000 Phrases	5000 Phrases	10,000 Phrases
Omega-T	29	29.5	29.7
Apertium Translator	29.2	29.9	30
CT translator (400,000 phrases)	33	34	34.5
CT translator (800,000 phrases)	34	35	35.8
CT translator (1,200,000 phrases)	36	36.3	36.7
CT translator (1,600,000 phrases)	40	41.2	43
CT translator (2,000,000 phrases)	42	43.2	44.4

Table 6. Automated evaluation using F_{mean} Score, for the English-French corpora.

Translation System	Average BLEU Score		
	1000 Phrases	5000 Phrases	10,000 Phrases
Omega-T	31.5	32.3	34
Apertium Translator	33.7	33.8	34.3
CT translator (100,000 phrases)	34.3	34.7	35
CT translator (300,000 phrases)	35.2	35.1	36.3
CT translator (500,000 phrases)	37	37.4	38

For English-French translation, the proposed CT system was demonstrated to be better than both Omega-T and Apertium in quality of translation for all corpus sizes. The results are shown in Tables 5 and 6 for English-Arabic and English-French translations respectively.

5.5. Summary

Results of the experiments indicate that the computational cost required to process a translation request is logarithmic to the count of the distinct phrases in the bilingual corpus (and, thus the size of the Corpus-Trie). Only a small fraction of CT nodes (5% to 20% percent of the log of the number of the nodes) have to be visited. A Corpus-Trie of two hundred million phrases has a worst-case response time of 27.57542 nodes. Responding to the translation request using Apriori-based algorithms would be much more expensive.

5.6. Limitations and Future Extensions

We devised a qualitative assessment to track the limitations of our system to detect false negative, which means that the translation could be extracted from the bilingual corpora but was not done by our CT system.

We built a testing sample for our qualitative assessment, the sample consisted of 1000 phrases, 60% of the phrases are included in the source language of the CT, 10% of the phrases are included partially in the CT, while 10% of the phrases are included in the CT but as fragments not the whole phrases continuously. Another 10% of the phrases were included but with synonyms of some of the words. The last 10 % of the phrases are not included at all.

The qualitative assessment is summarized by showing example of true positive and false negative in Table 7. We used English to Arabic CT system as we are fluent in both languages.

Table 7. The qualitative assessment.

Phrases	CT System Output	Comment
Phrases are included fully in the CT “Strabismus is a medical condition that is defined as the lack of coordination between the eyes”.	الحول هو حالة طبية تُعرّف على أنها نقص التنسيق بين العينين .	Good translation
Phrases are included partially in the CT “When Strabismus is detected at an older age, the chances of curing it are slimmer”.	No translation is Found	Although the following phrases are found in the CT but fragmented at different nodes and not included all in one phrase trie: a. When Strabismus is detected; b. at an older age; c. the chances of curing it; d. are slimmer.
Phrases are included but with different synonyms in the CT “Strabismus is a medical condition that is known as the lack of coordination between the eyes”.	No translation is Found	We got no translation because of synonyms not included in the CT

In our system, we don't have the notion of false negative as it only translates sentences that are presented in the Corpus Trie either as a whole sentence or as ordered subset of a source sentence in the Corpus Trie. Therefore, we can conclude from the qualitative assessment that one of the limitations of our system is that we have no mechanism to union translations for phrases fragments that are already included in our corpus. A minor limitation is the lack of synonyms in the phrases of the source language, which can be included easily.

6. Conclusions

In this paper, we have introduced new concepts in machine translation paradigms, examining a bilingual corpus by submitting a translation request including the phrase *S* in the source language. We treated the corpus as a database of frequent word sets. We proposed a data structure called a Corpus-Trie that compresses a bilingual parallel corpus into a compact data structure representing a frequent data items set. We presented all required algorithms using the trie to answer translation requests, with novel properties and exhaustive experiments. Experiments were performed on English-to-Arabic and English-to-French translations, although the proposed method is not restricted to any specific language. Moreover, the proposed Corpus-Trie can be extended from a bilingual corpus to accommodate multi-language corpora in future iterations. We included the following algorithms that implement the following:

- Building a phrase trie with translation, alignment and frequencies;
- Z-tree insertion: inserting a translation, alignment, and update frequency if available;
- Building a Corpus-Trie from several phrase tries;
- Inserting a new phrase in the Corpus-Trie;
- Searching for a phrase in the Corpus-Trie and retrieving its translation.

Future extensions will include:

- Generalization to multi-language translation, enabling a source language to have translations from multiple target languages in the same Corpus-Trie;
- Inverted structure of the Corpus-Trie to benefit two-way translation;

- Compact implementation of Corpus-Trie structure.

Author Contributions: Conceptualization, H.A.H.M. and H.A.M.; methodology, H.A.H.M. and H.A.M.; software, H.A.H.M.; validation, H.A.H.M. and H.A.M.; formal analysis, H.A.H.M. and H.A.M.; investigation, H.A.H.M. and H.A.M.; resources, H.A.H.M. and H.A.M.; data curation, H.A.H.M.; writing—original draft preparation, H.A.H.M.; writing—review and editing, H.A.H.M. and H.A.M.; visualization, H.A.H.M. and H.A.M.; supervision, H.A.M.; project administration, H.A.M.; funding acquisition, H.A.H.M. and H.A.M. Both authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-track Research Funding Program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

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

References

1. Liu, X.; Wong, D.F.; Chao, L.S.; Liu, Y. Latent Attribute Based Hierarchical Decoder for Neural Machine Translation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 2103–2112. [[CrossRef](#)]
2. He, W.; He, Z.; Wu, H.; Wang, H. Improved neural machine translation with SMT features. In Proceedings of the 30th AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 151–157.
3. Basmatkar, P.; Holani, H.; Kaushal, S. Survey on Neural Machine Translation for multilingual translation system. In Proceedings of the 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; pp. 443–448.
4. Wibawa, A.P. Invited Speech 2: An Overview of Machine Translation. In Proceedings of the 2nd International Conference of Computer and Informatics Engineering (IC2IE), Banyuwangi, Indonesia, 10–11 September 2019; p. 1.
5. Zhang, J.; Matsumoto, T. Character Decomposition for Japanese-Chinese Character-Level Neural Machine Translation. In Proceedings of the International Conference on Asian Language Processing (IALP), Shanghai, China, 15–17 November 2019; pp. 35–40.
6. Nakamura, N.; Isahara, H. Effect of linguistic information in neural machine translation. In Proceedings of the International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA), Denpasar, Indonesia, 16–18 August 2017; pp. 1–6.
7. Mukta, P.; Mamun, A.; Basak, C.; Nahar, S.; Arif, M.F.H. A Phrase-Based Machine Translation from English to Bangla Using Rule-Based Approach. In Proceedings of the International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, 7–9 February 2019; pp. 1–5.
8. Nahar, S.; Huda, M.N.; Nur-E-Arefin, M.; Rahman, M.M. Evaluation of machine translation approaches to translate English to Bengali. In Proceedings of the 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 22–24 December 2017; pp. 1–5.
9. Yashothara, S.; Uthayasanker, R.T.; Jayasena, S. Improving Phrase-Based Statistical Machine Translation with Pre-processing Techniques. In Proceedings of the International Conference on Asian Language Processing (IALP), Bandung, Indonesia, 15–17 November 2018; pp. 322–327.
10. Nair, T.; Idicula, S.M. Syntactic Based Machine Translation from English to Malayalam. In Proceedings of the International Conference on Data Science & Engineering (ICDSE), Cochin, Kerala, India, 18–20 July 2012; pp. 198–202.
11. Wang, X. Study of data mining based on Apriori algorithm. In Proceedings of the 2nd International Conference on Software Technology and Engineering, San Juan, PR, USA, 3–5 October 2010; pp. V2-400–V2-403.
12. Pavitra-Bai, S.; Ravi-Kumar, G.K. Efficient Incremental Item set Tree for approximate Frequent Item set mining on Data Stream. In Proceedings of the 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Bangalore, India, 21–23 July 2016; pp. 239–242.
13. Adhikary, A.; Ahmed, S. CorpMate: A framework for building linguistic corpora from the web. In Proceedings of the 19th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 18–20 December 2016; pp. 367–370.
14. Xiao, R.; Indurkha, N.; Damerau, F.J. Corpus creation. In *Handbook of Natural Language Processing*; CRC Press, Taylor and Francis Group: Boca Raton, FL, USA, 2010; ISBN 978-1420085921.
15. Arnold, D.J.; Balkan, L.; Meijer, S.; Humphreys, R.L.; Sadler, L. *Machine Translation: An Introductory Guide*; Blackwells-NCC: London, UK, 1994.

16. Samantaray, S.D. A Data mining approach for resolving cases of Multiple Parsing in Machine Aided Translation of Indian Languages. In Proceedings of the Fourth International Conference on Information Technology (ITNG'07), Las Vegas, NV, USA, 2–4 April 2007; pp. 401–405.
17. Wu, X.; Zhu, X.; Wu, G.; Ding, W. Data mining with big data. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 97–107.
18. Mahmud, M.S.; Huang, J.Z.; Salloum, S.; Emara, T.Z.; Sadatdiynov, K. A survey of data partitioning and sampling methods to support big data analysis. *Big Data Min. Anal.* **2020**, *3*, 85–101. [[CrossRef](#)]
19. Chen, M.; Han, J.; Yu, P.S. Data mining: An overview from a database perspective. *IEEE Trans. Knowl. Data Eng.* **1996**, *8*, 866–883. [[CrossRef](#)]
20. Qiu, H.; Gu, R.; Yuan, C.; Huang, Y. YAFIM: A Parallel Frequent Item set Mining Algorithm with Spark. In Proceedings of the IEEE International Parallel & Distributed Processing Symposium Workshops, Phoenix, AZ, USA, 19–23 May 2014; pp. 1664–1671.
21. Zhang, Z.; Zhao, H.; Ling, K.; Li, J.; Li, Z.; He, S.; Fu, G. Effective Subword Segmentation for Text Comprehension. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 1664–1674. [[CrossRef](#)]
22. Niu, X.; Qian, M.; Wu, C.; Hou, A. On a Parallel Spark Workflow for Frequent Itemset Mining Based on Array Prefix-Tree. In Proceedings of the IEEE/ACM Workflows in Support of Large-Scale Science (WORKS), Denver, CO, USA, 17 November 2019; pp. 50–59.
23. Sohrabi, M.K.; Barforoush, A.A. Parallel frequent itemset mining using systolic arrays. *Knowl. Based Syst.* **2013**, *37*, 462–471. [[CrossRef](#)]
24. Xun, Y.; Zhang, J.; Qin, X. Fidoop: Parallel mining of frequent itemsets using mapreduce. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 313–325. [[CrossRef](#)]
25. Rahman, M.; Rigby, P.; Palani, D.; Nguyen, T. Cleaning Stack Overflow for Machine Translation. In Proceedings of the IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), Montreal, QC, Canada, 26–27 May 2019; pp. 79–83.
26. Sazzed, S.; Jayarathna, S. A Sentiment Classification in Bengali and Machine Translated English Corpus. In Proceedings of the IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), Los Angeles, CA, USA, 30 July–1 August 2019; pp. 107–114.
27. Li, J.J.; Rossikova, Y.; Morreale, P. Natural Language Translator Correctness Prediction. *J. Comp. Sci. Appl. Inform. Technol.* **2016**, *1*, 11. [[CrossRef](#)]
28. Wang, X.; Tu, Z.; Zhang, M. Incorporating Statistical Machine Translation Word Knowledge into Neural Machine Translation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 2255–2266. [[CrossRef](#)]
29. España-Bonet, C.; Varga, Á.C.; Barrón-Cedeño, A.; Genabith, J.V. An Empirical Analysis of NMT-Derived Interlingual Embeddings and Their Use in Parallel Sentence Identification. *IEEE J. Sel. Top. Signal Process.* **2017**, *11*, 1340–1350. [[CrossRef](#)]
30. Nguyen, A.T.; Rigby, P.C.; Nguyen, T.V.; Karanfil, M.; Nguyen, T.N. Statistical translation of English texts to API code templates. In Proceedings of the IEEE International Conference on Software Maintenance and Evolution (ICSME), Madrid, Spain, 23–29 September 2018; pp. 194–205.
31. Banik, D.; Ekbal, A.; Bhattacharyya, P. Machine Learning Based Optimized Pruning Approach for Decoding in Statistical Machine Translation. *IEEE Access* **2019**, *7*, 1736–1751. [[CrossRef](#)]
32. Agić, Ž.; Vulić, I. JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 3204–3210.
33. Hettige, B.; Karunananda, A.S.; Rzevski, G. Phrase-level English to Sinhala machine translation with multi-agent approach. In Proceedings of the IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 15–16 December 2017; pp. 1–6.
34. Ji, B.; Zhang, Z.; Duan, X.; Zhang, M.; Chen, B.; Luo, W. Cross-lingual Pre-training Based Transfer for Zero-shot Neural Machine Translation. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
35. Liu, D.; Ma, N.; Yang, F.; Yang, X. A Survey of Low Resource Neural Machine Translation. In Proceedings of the 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Hohhot, China, 24–26 October 2019; pp. 39–393.
36. Shukla, M.B.; Chavada, B. A Comparative Study and Analysis of Evaluation Matrices in Machine Translation. In Proceedings of the 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 13–15 March 2019; pp. 1236–1239.
37. Kit, C.; Pan, H.; Webster, J. Example-Based Machine Translation: A New Paradigm. In *Translation and Information Technology*; Chinese University of Hong Kong Press: Hong Kong, China, 2002; pp. 57–78.
38. Ayu, M.A.; Mantoro, T. An Example-Based Machine Translation approach for Bahasa Indonesia to English: An experiment using MOSES. In Proceedings of the IEEE Symposium on Industrial Electronics and Applications, Langkawi, Malaysia, 25–28 March 2011; pp. 570–573. [[CrossRef](#)]
39. Ashraf, N.; Ahmad, M. Maximum Likelihood Estimation using Word Based Statistical Machine Translation. In Proceedings of the International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 17–19 July 2019; pp. 1919–1923. [[CrossRef](#)]
40. Xing, J.; Wong, D.F.; Chao, L.S.; V-Leal, A.L.; Schmalz, M.; Lu, C. Syntaxtree aligner: A web-based parallel tree alignment toolkit. In Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Jeju, Korea, 10–13 July 2016; pp. 37–42. [[CrossRef](#)]

41. Gale, W.A.; Kenneth, W.C. A Program for Aligning Sentences in Bilingual Corpora. *Comput. Linguist.* **1993**, *19*, 75–102.
42. Hoang, H.; Bogoychev, N.; Schwartz, L.; Junczys-Dowmunt, M. Fast Scalable Phrase-Based SMT Decoding. 2016. Available online: <https://arxiv.org/abs/1610.04265> (accessed on 12 August 2020).
43. Kunchukuttan, A.; Bhattacharyya, P. Faster Decoding for Subword Level Phrase-Based SMT between Related Languages. 2016. Available online: <https://arxiv.org/abs/1611.00354> (accessed on 12 August 2020).
44. Ziemiński, M.; Junczys-Dowmunt, M.; Pouliquen, B. The United Nations Parallel Corpus v1.0. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 3530–3534.
45. Omega-T: Open Source Translation Machine. Available online: <https://omegat.org/> (accessed on 12 August 2020).
46. Apertium: Open Source Translation Machine. Available online: <https://apertium.org/index.eng.html?dir=eng-epo#translation> (accessed on 12 August 2020).
47. Malik, P.; Baghel, A.S. A Summary and Comparative Study of Different Metrics for Machine Translation Evaluation. In Proceedings of the 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 11 January 2018; pp. 55–60. [CrossRef]
48. Lavie, A.; Sagae, K.; Jayaraman, S. The Significance of Recall in Automatic Metrics for MT Evaluation. In *Machine Translation: From Real Users to Research*; Frederking, R.E., Taylor, K.B., Eds.; AMTA 2004; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3265. [CrossRef]