A framework for phrase based SMT in the technical translation domain

Hisham Farouk Khodeir
A Framework For Phrase Based SMT
In The Technical Translation Domain

A Thesis Submitted to
the Department of Computer Science
in partial fulfillment of the requirement of
the degree of Master of Science

By
Hisham Farouk Khodeir
B.Sc.in Computer Science

Under the Supervision of
Professor Ahmed Rafea

May 2004
The American University in Cairo

A Framework For Phrase Based SMT

In The Technical Translation Domain

A Thesis Submitted By Hisham Farouk Khodeir
to the Department of Computer Science

May/2004

in partial fulfillment of the requirements for

the degree of Master of Science

has been approved by

Dr. Ahmed Rafea
Thesis Committee Chair / Advisor
Affiliation: The American University in Cairo.

Dr. Amr Goneid
Thesis Committee Reader / Examiner
Affiliation: The American University in Cairo.

Dr. Ashraf Abdelbar
Thesis Committee Reader / Examiner
Affiliation: The American University in Cairo.

Dr. Hisham ElShishiny
Thesis Committee Reader / Examiner

Affiliation: IBM Corporation - Egypt

Dr. Mikhail N. Mikhail
Chair of Computer Science

Dr. Fadel Assabghy
Dean of Sciences and Engineering
Acknowledgement

Many thanks to my supervisor Dr. Ahmed Rafea for his guidance, constructive criticism and support during this work and during my undergraduate life!

I would, also, like to thank Dr. Amr Goneid and Dr. Ashraf Abdelbar, my thesis committee, for all their valuable suggestions and feedback.

I would like to thank Dr. Hisham ElShishiny, my thesis external examiner, for taking the time to read every single page of my thesis and for his valuable suggestions and feedback on the work done.

My gratitude must go to my friends Amir Gouda and Mohamed Elbadrawy at Future Group their help and support are beyond words of gratitude.

Finally, my appreciation must go to my parents, wife, and kids who made it possible for me to live through the experience and gave me their support and help in countless ways. This thesis belongs to them!
Abstract

Machine translation is not commonly used in the technical translation domain. This is because this domain needs accurate consistent translation based on standard terminologies. Machine translation systems output is not precise and is used only by a user who needs just to understand what the source text is talking about in general.

In the late 90's machine translation research community has reported better results than available commercial MT systems found in the market by using a new approach called statistical machine translation. This approach is language independent and needs no hand crafted linguistic rules. This thesis is interested in researching the statistical machine translation approach and trying to apply it to the problem of the technical translation domain.

We proposed a statistical machine translation system and our experiments using a small corpus of size 20,000 sentences suggested that this system outperforms the well-established word based statistical machine translation system. In a small experiment, we showed that the output of the proposed system is better than the suggestions CAT tools supply to the human translator, and we suggest a new architecture to replace the fuzzy match suggestions found in the available commercial CAT tools.

In future work we need to do more experimentation using the new suggested architecture for the CAT tools. Also we need to enhance the statistical model used in our machine translation system by adding a syntax language model and to experiment the effect of this language model on the performance of the system.
Table of Contents

List of Tables ...........................................................................................................vii

List of Figures .......................................................................................................... viii

1. Introduction........................................................................................................... 1

  1.1 Machine Translation History ................................................................. 1
  1.2 Machine Translation Strategies .......................................................... 2
  1.3 Machine Translation Approaches ......................................................... 4
    1.3.1 Rule based or Knowledge base MT (RBMT/KBMT) .......... 4
    1.3.2 Corpus Based MT Systems ......................................................... 6
      1.3.2.1 Translation Memory (TM) .............................................. 6
      1.3.2.2 Structural Example based systems: ......................... 8
  1.4 Motivation ................................................................................................. 11
  1.5 Thesis objectives ..................................................................................... 11
  1.6 Thesis Structure ..................................................................................... 12

2 Statistical based Translation systems: ......................................................... 14

  2.1 Language Model .................................................................................. 16
  2.2 Translation model .............................................................................. 17
    2.2.1 Word based statistical translation models .......................... 18
    2.2.2 Phrase based Statistical Machine Translation models .... 21
  2.3 Decoder for statistical machine translation ........................................ 24
    2.3.1 Stack Decoding ........................................................................ 26
    2.3.2 Fast Greedy decoding .............................................................. 27
    2.3.3 Phrase based decoder .............................................................. 31

3. A Proposed SMT ........................................................................................ 32

  3.1 Training Corpus collection .................................................................. 32
  3.2 Generic SMT system components ...................................................... 33
  3.3 The proposed system components ...................................................... 34
    3.3.1 Alignment Matrix generation ............................................... 35
    3.3.2 Bi-directional Alignment Matrix generation ..................... 36
    3.3.3 Phrase alignment generation model .................................... 36
    3.3.4 Phrase Based decoder .......................................................... 40

4 Experimentation ............................................................................................ 45

  4.1 Evaluation Criteria ............................................................................ 45
  4.2 Experiment 1: IBM Model 4 versus Phrase alignment model .......... 46
    4.2.1 Experiment Objective ............................................................. 46
    4.2.2 Experiment Details ................................................................. 46
    4.2.3 Results analysis ..................................................................... 48
  4.3 Experiment 2: Phrase based alignment heuristics ......................... 51
List of Tables

Table 3.1: Corpus statistics.................................................................33
Table 3.2: Example of extracted aligned phrases ................................38
Table 4.1: GIZA++ training iterations .............................................46
Table 4.2: Experiment 1 results .........................................................48
Table 4.3: Experiment 2 results .........................................................52
Table 4.5: TM fuzzy match ..............................................................55
Table 4.6: Edit distance between SMT & Human translation ...............56
Table 4.7: TM suggestions versus human translation output ...............56
Table 4.8: CAT Vs. MT Useful suggestion .........................................57
List of Figures

Figure 1.1: The Vauquois triangle 1 .................................................................3
Figure 1.2: RBMT/KBMT symbolic represent 1 ..............................................4
Figure 1.3: Frame hierarchy example 1 ...........................................................5
Figure 1.4: Translation memory examples ......................................................7
Figure 1.5: Fuzzy Matching in TM .................................................................8
Figure 1.6: Example database in an EBMT .....................................................9

Figure 2.1 : Noisy channel model .................................................................15
Figure 2.2: IBM Model 1 ..............................................................................19
Figure 2.3: IBM Model 3 (from [Al-Onaiza and Knight,1998]) ......................20
Figure 2.4: Model 4 distortions .....................................................................20
Figure 2.5: Phrase extraction heuristics ..........................................................23
Figure 2.6: The minimum set problem [Kinght, 1999b] .................................24
Figure 2.7: Word reordering seen as TSP [Kinght, 1999b] ..............................25
Figure 2.8: Fast Greedy decoder illustration [Germann et al 2001] ...............30

Figure 3.1: SMT architecture based on Bayes’ descision rule .........................33
Figure 3.2: The proposed system architecture ................................................35
Figure 3.3: Alignment Matrix .......................................................................36
Figure 3.4 : Wrong alignment example .........................................................39
Figure 3.5: Search tree for traversing all possible n-grams in sentence ............42

Figure 4.1: Translation examples from experiment 1 .....................................50
Figure 4.2: SMT within CAT framework .......................................................58
1. **Introduction**

Machine translation is the automatic translation of text or speech from one language to another. It is one of the most important applications of NLP. The dream of building machines that let people from different cultures talk to each other easily is one of the most important goals of the NLP community. Unfortunately, MT is a hard problem. It is true that nowadays you can buy inexpensive packages that call themselves translation programs. They produce low-quality translations, which are sufficient for people who know enough about a foreign language to be able to understand the source with the help of a buggy translation.

The goal of many NLP researchers is instead to produce close to error free output that reads fluently in the target language. Existing systems are far from this goal for all but the most restricted domains like weather reports.

1.1 **Machine Translation History**

The idea of using computers to translate or help translate human languages is almost as old as the computer itself. Indeed, MT is one of the oldest non-numeric applications of computers [Hutchins, J., 1995].

Early proposals for the use of numerical techniques in MT can be traced back at least to 1947, when computers had just been successfully employed in deciphering encryption methods during the Second World War. A memo from Warren Weaver proposed specific strategies for using computers to translate natural languages [Weaver, 1955]. This memo initiated MT research in the USA and the rest of the world, with the first public demonstration of a Russian-English prototype MT system in 1954.
In the 1970’s, continued effort in MT yielded operational systems as Systran which was a Russian-English translation system and another system called Meteo began translating weather reports in 1976 [Arturo, T, 1999].

In the late 80’s and early 90’s a number of companies, especially large Japanese electronics manufacturers, began to market MT software for workstations. A number of products appeared for personal computers, and various Machine-Aided-Translation (MAT) tools such as translation memory began to be commonly used. This period also saw the emergence of work on speech translation and of statistical approaches to machine translation [Arturo, T, 1999].

Late 1990’s we are seeing powerful translation engines on personal computers, translation on the Internet, widespread use of translation memory and translator’s workbenches, multimedia and software localization, as well as increased interest in Corpus Based MT systems as the statistical machine translation approach [Arturo, T, 1999].

1.2 Machine Translation Strategies
MT systems are normally classified in terms of their basic strategy for carrying out translation [Arturo, T, 1999]. There are three main strategies:

**Direct:** Direct systems involve extensive string pattern matching, with some re-arrangement of the target string for conformance to the TL word order. Many early systems, as well as some recent MT software for personal computers employ this strategy.

**Transfer systems:** Transfer systems involve analysis of the source input into a transfer structure which abstracts away from many of the grammatical details of the SL. The idea is to facilitate translation by generalizing over different constructions.
After analysis, the SL structure is transferred into a corresponding TL structure which is then used to generate a TL sentence. Various types of transfer system may be identified, depending on the level at which transfer takes place. In general the more abstract the transfer representation, the easier it is to build the appropriate transfer module.

**Interlingua:** In interlingua systems SL sentences are analysed into a language neutral representation from which generation of TL sentences takes place, possibly after some language-independent manipulation of the interlingua representation. This strategy eliminates the need for a transfer step altogether.

These notions are illustrated using the Vauquois triangle shown in Fig. 1.1. The triangle illustrates in the vertical direction the amount of effort necessary for analysis/generation and in the horizontal dimension the amount of effort needed for
transfer. At the apex, transfer effort is minimum, while analysis and generation are at a maximum.

Obviously, this is a highly idealized view of MT, but it illustrates the point quite neatly. Variations on a basic strategy are possible. For example, a system may use, a hybrid of interlingua and transfer elements. There are also combinations of the basic direct and transfer strategies using statistical and other corpus-based techniques.

1.3 Machine Translation Approaches

We can divide the machine translation implementation approaches into two broad ways

(1) Rule-based MT

(2) Corpus-based MT.

1.3.1 Rule based or Knowledge base MT (RBMT/KBMT)

RBMT is characterized by a heavy emphasis on functionally complete understanding of the meaning of the source text prior to translation to the target text. RBMT/KBMT does not require total understanding, but assumes an interpretation engine can achieve successful translation into several languages. Most RBMT/KBMT is implemented based on the interlingual architecture [Seasly, 2003].

Figure 1.2: RBMT/KBMT symbolic representation
RBMT/KBMT systems must be supported by word knowledge and by linguistic semantic knowledge about meanings of words and their combinations. Thus, a specific language is needed to represent the meanings of sentences. In many RBMT/KBMT systems, knowledge is represented by frames that have named slots or features and values as shown in figure 1.3.

```
[ instance_of: save
  isa: physical_event
  id: save_1
  agent: user
  patient: [ instance_of: document
    isa: separable_entity
    id: document_1
    reference: definite ]
]
```

![Diagram of frame hierarchy example](image)

**Figure 1.3: Frame hierarchy example [Seasly, 2003]**

Once the source language is analyzed, it will be run through the augmenter. The augmenter is the knowledge base that converts the source representation into an appropriate target representation before synthesis into the target sentence.

RBMT/KBMT systems provide high-quality translations. However, they are quite
expensive to produce due to the large amount of knowledge needed to accurately represent sentences in different languages.

1.3.2 Corpus Based MT Systems:

The construction of ‘traditional’ rule based (RBMT) or knowledge-based MT systems (KBMT) is a lengthy, laborious and error-prone process. It is difficult to produce hand-crafted transfer rules to cover a wide variety of input. Frequently, when new rules are added rule conflict can produce unpredictable side effects. In addition, there is no well-known linguistic theory of transfer according to [Melby, 1986]. This led to the idea of translation by analogy principle explained in [Nagao, 1984] and [Sato et al, 1990], which makes use of a set of previously translated sentences (bilingual corpus) as opposed to the construction of hand-crafted monolingual grammars, bilingual lexicons and transfer rules. Since then, there has been an explosion of interest in approaches that use a bilingual corpus as the principal bilingual knowledge source. Such approaches use subtly different techniques and consequently take names to reflect this such as Example based Machine Translation (EBMT), Statistical Machine Translation (SMT) and Translation memory.

1.3.2.1 Translation Memory (TM)

A translation memory is a type of translation support tool and not really a machine translation system. The TM maintains a database of source and target language sentence pairs and whenever it finds a sentence in the text to be translated that exactly matches one of the sentence pairs in the database it automatically retrieves the equivalent target sentence and translates this sentence for the translator.
Translation memory is undeniably useful for the translation of certain types of repetitive documents but this technology as it is right now can only exploit a small portion of the knowledge residing in translators’ past production.

A first question that may be raised about this technology is what exactly is meant by an exact match. What qualifies as an exact match between a new source language (SL) segment and the contents of the TM database? In most commercial systems the notion of similarity is based on the number of shared characters or what we can call generally the ‘edit distance’ between strings. So if we have as in figure 1.4 a sentence (1) to be translated and we got in our TM database sentences (2) and (3) the translation memory will conclude that sentence (1) is matching sentence (2) more than (3) since (2) differs from (1) by only 4 characters although the correct answer should be the reverse [Macklovitch et. al, 2000].

(1) The wild child is destroying his new toy.

(2) The wild chief is destroying his new tool.

(3) The wild children are destroying their new toy.

Figure 1.4: Translation memory examples

From the previous example, we can conclude that this TM technology could be only useful in translation tasks such as document revisions or updates and perhaps certain types of technical maintenance manuals. Most translators find that this technology is of much help to them but also they are convinced that their archives actually contain much useful information on a sub-sentence level that is not being exploited by these systems.

TM systems are unable to back off and retrieve examples of phrases even though such units may well be present in the database. Suppose that example (4) in figure 1.5 is a new input sentence made up of twenty words, each with the same
length The TM database contains no exact match for (4) but does contain the SL sentence (5). Notice that the two sentences share an identical sub string w1 … w5 which in both cases is marked off from the rest of the sentence by a comma. However, since this sub string contains only 25% of the sentence’s total number of characters, it is doubtful that any current TM system would be able to retrieve it among its fuzzy matches [Macklovitch, 2000].

(4) w1 w2 w3 w4 w5, w6 … w20
(5) w1 w2 w3 w4 w5,w21 … w35

Figure 1.5: Fuzzy Matching in TM

1.3.2.2 Structural Example based systems:
EBMT is often linked with the related technique of “Translation Memory” (TM). Some researchers regard EBMT and TM as the same thing, while others believe there is a main difference between the two, rather like the difference between computer-aided translation and MT. Although they have in common the idea of reuse of examples of already existing translations, they differ in that; TM is an interactive tool for the human translator, while EBMT is an essentially automatic translation technique or methodology. They share the common problems of storing and accessing a large corpus of examples, and of matching an input phrase or sentence against this corpus; but having located a (set of) relevant example(s), the TM leaves it to the human to decide what, if anything, to do next, whereas this is only the start of the process for EBMT.

The basic idea in structured EBMT is simply to translate a sentence it uses previous translation examples of similar sentences. The assumption being that many translations are simple modifications of previous translations. A fully fledged EBMT
system retrieve more than one example, identify fragments which match parts of the input sentence and combine these fragments into a TL sentence [Somers, 1999].

A typical EBMT system consists of the following main components:

1. An example database of aligned source and target sentences. Translation examples are fully annotated tree structures with alignments at the lexical and structural level. These aligned tree structures serve as the rule base against which parsed SL input sentences are matched. Normally the dependency structure of example sentences may be obtained through manual annotation, or via a parser. Each example includes sub-sentential alignments indicating which fragments between the source and target are in translation correspondence see figure (1.6) below. Again these alignments are made manually or through (semi) automated means using bilingual dictionaries or word and term alignment algorithms.

![Diagram of EBMT example database]

Figure 1.6: Example database in an EBMT [Somers, 1999]
2. A matching algorithm that identifies the examples that most closely resemble all or part of the input sentence. Typically, the closest matching SL structure to the parsed SL input is retrieved. The alignments at the lexical and structural level between translation examples enable the retrieval of translations of segments of the SL input from other translation examples in the corpus. Matching against a set of tree structures is a more complex task than matching against a set of raw translation examples and involves a considerable computational cost. Structural also requires a significant amount of external linguistic knowledge in the form of parsers and perhaps bilingual lexicons. This detracts from portability. However one advantage of including structural information in translation examples is the ability to represent explicitly alignments between languages that indicate a structural divergence

3. A combination algorithm that reconstructs the input sentence through a combination of fragments from the source side of the example sentences.

4. A transfer and composition algorithm that extracts corresponding target fragments and combines them into an appropriate TL sentence.

EBMT is an attractive approach to translation because it avoids the need for manually derived transfer rules. However, it requires analysis and generation modules to produce the dependency trees needed for the examples database and for analyzing the input sentence. Another problem with EBMT is computational efficiency, especially for large example databases, although parallel computation techniques can be applied to solve this problem.

The focus of this research work will be on using corpus-based approaches. These approaches rely on large amounts of bilingual corpora for carrying out translation. The corpus based MT systems tries to extract the linguistic knowledge
needed for translation from the corpus instead of hand crafted rules written by linguists. These approaches have gained a lot of focus during the recent years from the NLP community after reaching a plateau in the research that is dependent on traditional knowledge based systems.

1.4 Motivation
In the area of technical translation and product localization; even the industry leaders are still using only CAT tools in the process of translation of technical documentation, software user interface and online manuals. When we talk about the technical translation into Arabic, which is the focus of this work, MT is for sure not used at all. This is due to the fact that the output of machine translation doesn't help in increasing the productivity of the human translator when compared to the traditional computer aided translation tools which rely only on what is called translation memory technology which is a database of past translations for a specific product.

After the exciting work accomplished over the past decade in the field of Statistical Machine Translation (SMT), Is it still better for translators to use the CAT tools as it is found in the market nowadays or they can benefit more from embedding SMT within the process of localization?

1.5 Thesis objectives
This thesis aim is to answer the following questions:

1) Is it possible to use statistical MT within the framework of the technical translation jobs and give better results than the current CAT tools alone?

2) What is the good architecture of this system? Is it possible to replace the fuzzy logic matching module by the SMT module?

3) The corpus size will not be as large as the general translation domain if we divided the technical domain into more specific areas
such as mobiles, printers, general purpose software, automotive. So a 20,000 sentence corpus will be considered large corpus since it is difficult for a localization company to have access to bigger focused corpus. Will it be possible to get good results using such size of corpus?

4) Can we enhance the quality of the translation by adding simple heuristics in the extraction process of phrases from word alignments similar to Och et. al[1999]. Or this will not be useful in our application.

1.6 Thesis Structure
The thesis will be structured as follows; first chapter 2 will discuss the related work done in the area of interest of this thesis. In this chapter we will review statistical machine translation systems based on the source channel model by explaining different word based translation models, n-gram language models and also different algorithms used to build word based statistical decoders. We will also review phrase based statistical machine translation and give a review about approaches used in building the phrase translation model.

Chapter 3 will show the proposed system design and architecture. We will explain the different components that we used in our phrase based translation model and we will show the implementation details of these components together with an explanation of the phrase decoder developed and different decisions taken in the design of the system.

Chapter 4 will present the experimentation results obtained during this work showing how the proposed system outperformed the word based statistical models and
also how we can make use of the proposed system within the framework of technical translation environment that uses CAT tools.

Chapter 5 will reach the conclusions of this work together with possible paths that could be taken in future work to enhance the outcome of the proposed system.

Finally, we have included in the appendix samples taken from the output of the developed system together with main source code of the system.
2 Statistical based Translation systems:

The aim of this chapter is to provide an overview and background of the existing approaches to statistical based machine translation systems that are related to this thesis objective.

SMT system constructs a general model of the translation relation that lets the system acquire specific rules automatically from bilingual and monolingual text corpora. These rules are usually coarse and probabilistic. The most established SMT systems are based on word-for-word substitution. An advantage of the SMT approach is that designers can improve translation accuracy by modestly changing the underlying model rather than using large handcrafted resources. [Knight, 1999a]

The switch in the NLP research community from rule based systems to statistical and corpus-based systems started in the end of the 80’s and the early 90’s with the publication of a very influential paper by a group at IBM [Brown, 1993]. Their statistical model is called a noisy channel model. This model is widely used in signal processing to recover the original signal from a signal with noise.

This statistical model (the noisy channel model.) has two central statistical components a source generator and a transfer channel. A source signal generator generates a signal according to a statistical model and a transfer channel corrupts the signal according to a statistical channel model. The parameters of the models can be obtained from samples of the generated signal and the input output pairs of the transfer channel. Given the parameters of the statistical models, one can recover the original signal from an observed noisy signal, the output of the transfer channel.

This idea can be applied to statistical machine translation, such as a French to English translation system. In the noisy channel model, we use a well formed English
sentence generator and an English to French transfer channel. Hence, we consider that the channel corrupts an English sentence and modifies it to appear as French. The task of machine translation system is to recover the original English sentence from an observed French sentence.

![Diagram](image)

Figure 2.1: Noisy channel model []

Figure 2.1 illustrates this process. The upper left box generates a well formed English sentence $e$, with probability $P(e)$. This sentence is subsequently translated into a French sentence $f$ in the channel (the upper right box) with probability $P(f|e)$. The goal of translation (or decoding) is to recover the original $e$ given an observed sentence $f$. Any sentence $e$ can be a potential translation of $f$, but some are more probable than others. We aim to select the most probable $e$ which gives the highest $P(e|f)$, or using Bayes rule equivalently the highest $P(f|e) \cdot P(e)$.

A typical Statistical Based Machine Translation system consists of the following main components:

1) A **Language model** that is responsible for calculating the $P(e)$ so as to ensure that the outcome of the translation is syntactically correct.

2) A **Translation model** that is responsible for calculating the $P(f|e)$ we have to know that the translation model doesn’t necessarily turn $e$ into good $f$ since
part of this problem is the responsibility of the independently trained language model

3) A **decoder** which is responsible to translate an observed foreign sentence into the source sentence this is done by calculating argmax $P(e) \cdot P(f|e)$ for all possible $e$. This process is called decoding. It is impossible to search through all possible sentences, but it is possible to inspect a highly relevant subset of such sentences using heuristic search techniques. So what we get at the end is the most likely translation.

2.1 Language Model
Concerning the Language modeling we need to build a machine that assigns a probability $P(e)$ to each $e$ sentence. The statistical language model is based on this simple idea. Just record every sentence that anyone ever says in the language.

Suppose you record a database of one billion sentences. If the sentence “how's it going?” appears 76,413 time in that database then we say $P(\text{how's it going?}) = 76,413/1,000,000,000 = 0.000076413$. We can use the web to build the needed monolingual corpus. So by this a sentence as “I like snakes” is less probable than “I hate snakes”. The most widely used statistical language modeling is N-grams. In this model a string is broken down into components (substrings) and n word substring is called n-gram. If $n=2$ we say bigram, if $n=3$ we say trigram. If a string contains a lot of reasonable n-grams then maybe it is a reasonable string. I will explain the bigram language model by an example [Knight, 1999a] and this can be generalized to n-gram language models.

Let $b(y \mid x)$ be the probability that word $y$ follows word $x$. We can estimate this probability from online text. We simply divide the number of times we see the phrase “xy” by the number of times we see the word “x”. That's called a conditional bigram
probability. Each distinct $b(y \mid x)$ is called a parameter.

A commonly used n-gram estimator looks like this:

$$b(y \mid x) = \frac{\text{number-of-occurrences}("xy")}{\text{number-of-occurrences}("x")}$$

$P(\text{I like snakes that are not poisonous}) \sim$

- $b(\text{I} \mid \text{start-of-sentence})$
- $b(\text{like} \mid \text{I})$
- $b(\text{snakes} \mid \text{like})$

...$

- $b(\text{poisonous} \mid \text{not})$
- $b(\text{end-of-sentence} \mid \text{poisonous})$

In other words, what's the chance that you'll start a sentence with the word "I"? If you did say "I", what's the chance that you would say the word "like" immediately after? And if you did say "like", is "snakes" a reasonable next word? And so on. [Knight, 1999a]

2.2 Translation model

We can divide the statistical machine translation approach broadly into two main models depending on the algorithm used in the translation model and decoding components. The first one can be named as single-word based models (SWB). Models of this kind assume that an input word is generated by only one output word as [Brown, 1993]. This assumption does not correspond to the nature of natural language where in some cases we need to know a word group in order to obtain a correct translation. One initiative for overcoming the above restriction of single word models is known as the template based approach where still the underlying model is based on single word translation table. [Och, 1999] In this approach an entire group of adjacent
words in the source sentence may be aligned with an entire group of adjacent target
words in the target sentence. A template establishes the reordering between two
sequences of word classes.

Recent works in the area of statistical based translation presented what is
called the phrase-based statistical approaches [Koehn, 2003]. These methods
explicitly learn the probability of a sequence of words in a source sentence being
translated to another sequence of words in the target sentence.

2.2.1 Word based statistical translation models

Now I will give a brief description about the IBM translation models 1,2,3,4,5,
which are the well known word based models in the statistical machine translation
area. A full description could be found in [Brown, 1993] and their decoding algorithm
in [Berger, 1996].

Model 1 is the simplest model, and later models are extensions to previous
models. All models are word-based models. The input and output of the channel is
just sequences of words, and the channel operations are word duplication (including
insertion and deletion, word movements and word translations. To follow is a brief
description for each model [Knight, 1999a].

Fig 2.2 illustrates how Model 1 works. First the length of the target word
sequence is determined (the target in the noisy channel refers to the source in normal
translations) based on the source length. Next each output position is filled by copying
one of the source words. This copy operation works as word-duplication and word
movement.
Source word may be copied more than once. Some of the source words may not be copied to the target. This model and all subsequent models assume that a special word NULL exists in the source sentence, copying the NULL word acts as word insertion. The probability of copying a source word is assumed uniform. After the target positions are filled the words are translated independently according to a word translation table $t(\text{fe})$.

Model 2 extends Model 1 by employing a more general probability table for the copy operation rather than the uniform probability as in Model 1. This probability is conditioned on the source and target length $a(\text{sourcePosition}, \text{targetPosition}, \text{sourceLength}, \text{targetLength})$.

Model 3 introduces a new parameter called word fertilities, which controls the number of word duplication operation based on the source word see figure 2.3.
Figure 2.3: IBM Model 3 from [Yamada,2001]

In model 3 the word move operations probabilities are given by the table
d(targetPosition| SourcePosition, targetLength, sourceLength) which is called the
distortion table

In model 4, the distortion table is divided into two tables; d1 and d>1. Figure
2.4 shows how these tables are used. A source word is called fertile if its fertility is
greater than zero.

Figure 2.4: Model 4 distortions
In figure 2.4 the words e1 and e3 are fertile. The leftmost target position for a fertile word is called the head, and the average of the target positions for a fertile word is called the center. The target words f1 and f4 are the heads, and the center for e1 is the position where f2 is placed. The distortion table d1 is used to decide the position of heads, and the d>1 is used for non-heads. The d1 table specifies the offset from the center for the previous fertile word and is conditioned on the word classes of the previous fertile word and the target word. The word classes C() are automatically derived using a clustering algorithm [Brown, 1993]. The d>1 is similar but the offset is measured from the previous target position and is conditioned on the word class of the target word. Using offsets rather than absolute positions rewards whole phrasal movements.

Model 5 is basically the same as Model 4, except it adjusts the probabilities to avoid deficiency. The distortion probability table allows moving more than one source words into the same target position since each word move is independent from others so in Model 5 additional variables are introduced to represent vacant and valid target positions and to enforce a source word is copied only to a vacant and valid position.

2.2.2 Phrase based Statistical Machine Translation models

The principal innovation of the phrase based translation model is that it attempts to calculate the translation probabilities of word sequences rather than of only single words as IBM models [Koehn, 2003]. The other property of this translation model is that the alignment between phrases is one to one and continuous.

The generative process, which allows for the translation of a sentence, can be broken down into the following steps: First, the input sentence is segmented into phrases. Then each phrase is translated to the corresponding output phrase. The output
sentence is made by concatenating and reordering the output phrases to generate the target sentence.

During the last couple of years the statistical machine translation community has tried to enhance the translation output of IBM models by adding the concept of phrase to the translation model and also by adding lexical, syntax and semantic knowledge to the system. Some of the approaches could be found in [Marcu, 2002] and [Koehn, 2003]

The idea of phrase based SMT and adding external linguistic knowledge to the system was thought about to try to solve a number of challenges faced by the pure word-based statistical translation models as follows:

- The first problem is with multiple English words being translated from a single foreign word, which is not allowed by the IBM alignment scheme.

- The second is the translation of multiple word phrases which do not decompose easily into word for word translations because of non-compositional semantics.

- Finally, a practical problem in the estimation of the parameters of the IBM model is that only reordering local to an area of a few words can be estimated with any accuracy, making larger syntactic transformations difficult to capture.

It is very hard to address these challenges within the word based statistical machine translation framework. All of the parameters are tied to words, and these problems are tied to the behavior of groups of words which are called "chunks" and which might be described using linguistic vocabulary such as verb group or noun
phrases. The behavior of chunks is above the word level, and IBM model fails to capture this behavior.

Various researchers have improved the quality of statistical machine translation systems with the use of phrase or chunk machine translation. Most recently published methods [Zens, 2002], [Venugopal, 2003] and [Koehn, 2003] on extracting phrase translation tables from a parallel corpus start with a word alignment.

![Phrase extraction heuristics diagram](image)

**Figure 2.5: Phrase extraction heuristics**

The main idea is to align the parallel corpus bi-directional e.g. Spanish to English and English to Spanish as show in the above figure [2.5]. This generates two alignments that have to be reconciled. If we intersect the two alignments, we get high precision alignments with low recall and if we do the union, we get a high recall with low precision alignments. To extract phrases from these word alignments researchers consider any continuous block of words aligned to another continuous block in the
target words is considered a phrase. By this, they generate Phrase translation table to be used together with the word translation table.

2.3 Decoder for statistical machine translation

A good decoding algorithm is critical to the success of any statistical machine translation system. The decoder's job is to find the translation that is most likely according to a set of previously learned parameters. If we observe a new sentence $f$, then an optimal decoder will search for an $e$ that maximizes $P(e|f) \sim P(e) \cdot P(f|e)$

The decoding problem in statistical machine translation can be divided into two sub problems. The first is selecting a concise set of source words according to the $P(f|e)$ and the other is selecting a good source word order based on the $P(e)$. According to Knight [1999b] The first sub problem is like solving the minimum set coverage problem. See figure [2.6] And the other problem is like solving the traveling salesman problem. see figure [2.7]

![Diagram](image)

Figure 2.6: The minimum set problem [Kinght, 1999b]
Selecting a Concise Set of Source Words is Like Solving the Minimum Set Cover Problem. A channel model "Translation model" with overlapping one to many dictionary entries will typically license many decodings. The source model may prefer short decodings over long ones. Searching for a decoding of length \( \leq n \) is difficult resembling the problem of covering a finite set with a small collection of subsets. In the example shown above the smallest acceptable set of source words is \{ she and cooked however left comma period \}

![Diagram](image)

**Figure 2.7: Word reordering seen as TSP [Knight, 1999b]**

If we assume that the channel model offers deterministic word for word translations then the bigram source model takes responsibility for ordering them. Some word pairs in the source language may be illegal. In that case finding a legal word ordering is like finding a complete circuit in a graph. In the graph shown above a sample circuit is "boundary -> this -> year -> comma -> my -> birthday -> falls -> on -> a -> Thursday -> boundary." If word pairs have probabilities attached to them then word ordering resembles finding the least cost circuit also known as the Traveling Salesman Problem.
Since these two problems are proved to be NP-complete then to build a statistical machine translation it is possible to devise approximation algorithms like those devised for other NP Complete problems. So far statistical translation research has used heuristic beam search algorithms to solve these problems. To follow is a brief explanation for these algorithms.

2.3.1 Stack Decoding

The stack decoding (also called A*) decoding algorithm is a kind of best-first search which was first introduced in the domain of speech recognition [JeLinek, 2001]. The generic stack decoding algorithm is as follows:

1. Insert into the stack all the single-branch paths corresponding to the input string according to the translation lexicon. Arrange the entries in a descending order.

2. Take the top entry off the stack. If this entry is a complete translation then stop –this is the best path- else evaluate all possible next word extensions and insert them into the stack in a descending order.

3. Repeat the above step until the stopping criterion in step 2 is satisfied.

One crucial difference between the decoding process in speech recognition and machine translation is that speech is always produced in the same order as its transcription. Consequently, in speech recognition decoding there is always a simple left-to-right correspondence between input and output sequences. This change makes decoding significantly more complex in machine translation. Instead of knowing the order of the input in advance we must consider all $n!$ permutations of $n$-word input sentence.

Another important difference between speech recognition and MT decoding is the lack of reliable heuristics in MT. A heuristic is used to estimate the cost of
completing a partial hypothesis. A good heuristic makes it possible to accurately compare the value of different partial hypothesis and to focus the search in the most promising direction. The left to right restriction in speech recognition makes it possible to use a simple yet reliable heuristics which estimate cost based on the amount of input left to decode. Without a heuristic a classic stack decoder will almost always find that shorter hypothesis looks more attractive than longer ones since as we add more words we end up multiplying more and more terms to find the probability so longer hypothesis will be at the end of the stack. So to solve this issue Germann et. al [2001] used more than one stack to force hypothesis to complete fairly. They had one stack for each subset of input words. This way a hypothesis can only be pruned if there are other better hypothesis that represent the same portion of the input. At each iteration they choose one hypothesis from each stack to be extended.

The stack decoder for Model 3 builds the translation incrementally by applying operations to hypothesis. The decoder used four operations:

- **Add** adds a new English word and aligns a single French word to it.
- **AddZfert** adds two new English words. The first has fertility zero, while the second is aligned to a single French word.
- **Extend** aligns an additional French word to the most recent English word, increasing its fertility.
- **AddNull** aligns a French word to the English NULL element.

To reduce the cost of AddZfert they considered only certain English words as candidates for zero-fertility basically words which both occur frequently and have a high probability of being assigned null alignment. This was extracted from the training data. Second they only used zero fertility words if it increases the language model probability more than decreasing the alignment probability.

### 2.3.2 Fast Greedy decoding

Another alternative for solving many instances of NP-complete problems is the greedy method. Instead of deeply probing the search space, such greedy methods
typically start out with a random approximate solution and then try to improve it incrementally until a satisfactory solution is reached.

The greedy decoder developed by Germann et. Al [2001] starts the translation process from an English gloss of the French sentence given as input. The gloss is constructed by aligning each French word $f_j$ with its most likely English translation $e_f$ ($e_f = \text{argmax}_e(ef_j)$). For example, in translating the French sentence “Bien entendu, il parle de une belle victoire.”, the greedy decoder initially assumes that a good translation of it is “Well heard, it talking a beautiful victory” because the best translation of “bien” is “well”, the best translation of “entendu” is “heard”, and so on. Once the initial alignment is created, the greedy decoder tries to improve it by applying one of the following operations:

- **translateOneOrTwoWords($j_1,e_1,j_2,e_2$)** changes the translation of one or two French words, those located at positions $j_1$ and $j_2$ from $ef_j$ into $e_1$ and $e_2$. If $ef_j$ is a word of fertility 1 and $e_k$ is NULL, then $ef_j$ is deleted from the translation. If $ef_j$ is the NULL word, the word $e_k$ is inserted into the translation at the position that yields the alignment of highest probability. If $ef_j = e_1$ or $ef_j = e_2$, this operation amounts to changing the translation of a single word.

- **translateAndInsert($j,e_1,e_2$)** changes the translation of the French word located at position $j$ from $ef_j$ into $e_1$ and simultaneously inserts word $e_2$ at the position that yields the alignment of highest probability. Word $e_2$ is selected from an automatically derived list of 1024 words with high probability of having fertility 0. When $ef_j = e_1$ this operation amounts to inserting a word of fertility 0 into the alignment.

- **removeWordOfFertility0($i$)** deletes the word of fertility 0 at position $i$ in the current alignment.

- **swapSegments($i_1,i_2,j_1,j_2$)** creates a new alignment from the old one by swapping non-overlapping English word segments[$i_1,i_2$] and [$j_1,j_2$]. During the swap operation, all existing links between English and French words are preserved. The segments can be as small as a word or as long as $|e|-1$ words, where $|e|$ is the length of the English sentence.

- **joinWords($i_1,i_2$)** eliminates from the alignment the English word at position $i_1$ (or $i_2$) and links the French words generated by $e_1$ (or $e_2$) to $e_2$ (or $e_1$).
In a stepwise fashion, starting from the initial gloss, the greedy decoder iterates exhaustively over all alignments that are one operation away from the alignment under consideration. At every step, the decoder chooses the alignment of highest probability, until the probability of the current alignment can no longer be improved. When it starts from the gloss of the French sentence “Bien entendu, il parle de une belle victoire.”, for example, the greedy decoder alters the initial alignment incrementally as shown in Figure [2.8], eventually producing the translation “Quite naturally, he talks about a great victory.”. In the process, the decoder explores a total of 77421 distinct alignments/translations, of which “Quite naturally, he talks about a great victory.” has the highest probability.
The greedy decoder [Germann, 2001] is a viable alternative to the traditional stack decoding algorithm. Even when the greedy decoder used a set of operations that is optimized for speed in which at most one word is translated, moved or inserted at a time and at most 3 word long segments are swapped the translation accuracy is affected only slightly. In contrast, the translation speed increases at least one order in magnitude. We can consider the greedy decoder as a hill climbing algorithm for decoding.
2.3.3 Phrase based decoder

The phrase decoder that was developed and discussed in [Koehn, 2003] used the stack decoding discussed previously as its base algorithm. During decoding, the foreign sentence is segmented into a sequence of I phrases and each foreign phrase is translated into an English phrase. The English phrases may be reordered according to the relative distortion probability distribution \( d(a_i - b_{i-1}) \) where \( a_i \) denotes the start position of the foreign phrase that was translated into the \( i \)th English phrase and \( b_{i-1} \) denotes the end position of the foreign phrase translated into the \((i-1)\)th English phrase.

The phrase decoder is using Bayes rule \( P(e|f) = \arg \max_e p(f|e) \ p(e) \)

Where \( p(f|e) \) is decomposed into

\[
Pr((f_1^{i-1}|e_1^i) = \Pi_{i=1}^{i-1} p(f_i'|e_i') \ d(a_i - b_{i-1})
\]

In order to make the computation tractable they prune weak hypothesis from the stack based on the cost they incurred so far and a future cost estimate. Also they use a beam size of \( n \) best hypothesis up till now any hypothesis lower than this \( n \) beam size is removed from the stack.

The cost of a hypothesis is calculated based on the \( P(e|f) \) of this hypothesis up to this point and the future cost estimate which is calculated by first extracting all translation options (phrase translation) that is valid for the current input sentence then store these translation options together with their cost which is the phrase translation probability multiplied by the language model for this phrase only and ignoring the distortion probability \( d(a_i - b_{i-1}) \).
3. A Proposed SMT

We chose to use the SMT approach since it can learn language rules automatically and without the need of language experts. Moreover according to recently published results IBM SMT system outperformed classical rule based machine translation systems during a recently held MT competition.

In order to be able to have a functioning SMT system we need to have a training bilingual corpus and to have a translation model and accordingly develop an SMT system. Finally we should have a methodology for evaluating system output and report the results. In this chapter we will explain our different system components and the concept behind this design.

3.1 Training Corpus collection

Since this work is interested in the performance of statistical MT in the technical translation domain and in order to develop a statistical MT system we need to have a training corpus that should be from the same domain for which the MT system is used I was able to collect translation memories of previously translated technical documentation in the areas of mobile phone documentation and software, computer printers documentation and software, and automotive documentation. These TMs were translated from English to Arabic. The statistics of this corpus could be found in table 3.1 below.

In order to be able to use the translation memories we had to develop some macros to extract a sentence aligned corpus after removing as much as possible all the noise found in these TMs. This noise was due to the TM file formats, which include many tags that is related to the formatting of the text and has no relation with the translation itself.
<table>
<thead>
<tr>
<th>Domain</th>
<th>Sentences</th>
<th>Words</th>
<th>Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Phones</td>
<td>19148</td>
<td>149127</td>
<td>9558</td>
</tr>
<tr>
<td>Printers</td>
<td>19840</td>
<td>228555</td>
<td>13720</td>
</tr>
<tr>
<td>Automotive</td>
<td>18634</td>
<td>220205</td>
<td>13276</td>
</tr>
<tr>
<td>All (Joined corpus)</td>
<td>57622</td>
<td>780770</td>
<td>29709</td>
</tr>
</tbody>
</table>

Table 3.1: Corpus statistics

3.2 Generic SMT system components

One of this work objectives is to evaluate the performance of the word based SMT systems and take it as a baseline to compare it with the phrase based systems. We had to build the standard word based SMT systems. You can see the basic architecture for this type of systems in figure 3.1 below.

![SMT architecture based on Bayes' decision rule](image)

Figure 3.1: SMT architecture based on Bayes' decision rule

We can see from the generic architecture that we need to have several components to be able to have a running SMT systems based on the source channel approach. During the experimentation of this work, we used the GIZA++ freely

33
available toolkit which implements the lexicon and alignment model based on IBM models 1 through 4 discussed in chapter 2 of this thesis. GIZA++ is an extension to the program GIZA part of the SMT toolkit “EGYPT” developed by the statistical machine translation team of ISI/USC university during a summer workshop in 1999 at the center for language and speech processing at John Hopkins university [Al-Onaizan et al, 1999]

The GIZA++ was written and designed by Franz Och and can be downloaded from:

http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/GIZA++.html

For the decoding component, which is shown in figure 3.1 as the global search problem, we used the ISI Rewrite greedy decoder developed in ISI/USC [Germann, 2001] and discussed in chapter 2 of this thesis. This decoder works with the output files of the GIZA++ toolkit.

3.3 The proposed system components

In this work, we want to see how much can we enhance the SMT performance by adding phrase translation knowledge to the system.

Koehn et. al in [Koehn, 2003] suggest that the highest levels of performance can be obtained through relatively simple means: heuristic learning of phrase translations from word-based alignments and lexical weighting of phrase translations. Surprisingly, learning phrases longer than three words and learning phrases from high-accuracy word level alignment models does not have a strong impact on performance. Learning only syntactically motivated phrases degrades the performance of the proposed systems.” So we decided to use the simple phrase based method described by Franz Och et al.in [Och, 1999] and [Kohen, 2003] in order to evaluate how much the state of the art phrase based statistical machine translation can enhance
the productivity and help the human translators in the domain of technical translations. The proposed system architecture can be shown in figure 3.2

![Diagram of the proposed system architecture]

Figure 3.2: The proposed system architecture

### 3.3.1 Alignment Matrix generation

In order to train the phrase alignment model that will be used by the decoder later we had to develop the first component of our phrase based machine translation system, which was the alignment matrix generation shown in figure 3.2.

The alignment matrix generation tool uses the word alignment file generated by the GIZA++ toolkit as an input to generate what we call the matrix file that contains for every sentence aligned in the corpus a matrix showing the alignment. If a source word $e_i$ is aligned with a target word $f_j$ we place 1 in matrix position $[i,j]$ see figure 3.3

35
$ 1 0 0 0 0 0 0$

spark 0 0 0 1 1 0
plug 0 0 0 0 0 0

Figure 3.3: Alignment Matrix

tightening 0 0 1 0 0 0

3.3.2 Bi-directional Alignment Matrix generation

We implemented another module that could be used in building the word alignment matrix. This module is the implementation of the algorithm explained by Franz Och et al in [Och, 1999] as follows; after aligning a parallel corpus bi-directionally using GIZA++; construct the intersection matrix of the two word alignments generated and then add new alignment points that exist in the union of two word alignments if these points satisfies the following two constraints:

1. A new alignment point is added if it connects at least one previously unaligned word in the intersection.

2. And this new alignment point is directly adjacent to an already existing alignment point in the intersection.

3.3.3 Phrase alignment generation model

Before describing the algorithm used in the training of the phrase alignment model, we have first to define what a bilingually aligned phrase means. A bilingual phrase is defined as a pair of m consecutive source words that has been aligned with n
consecutive target words with the exception that any null aligned word will be included in the phrase.

In table 3.2 we can see the phrases extracted from the aligned sentence shown in figure 3.3. We have included in Appendix A of this thesis also part of the phrase table generated from the proposed system.

- For I = MaxsPhraseLen downto 1 Do
  - Init TotalNumPhrases //Total number of extracted phrases
  - For all aligned sentences in the Matrix file Do
    - For J = 0 to (SrcLen - I) Do //extract all possible phrases of len I
      - Init MinTrgtIdx & MaxTrgtIdx
      - For K = 1 to I Do //Loop over SrcPhrase
        - AddSrcIdx
        - GetAlignTrgt(K+J, MintrgtIdx, MaxTrgtIdx)
      - GetNullAlignedTrgt(MintrgtIdx, MaxtrgtIdx)
      - If ContinuousPhrase = True then
        - If SrcPhrase = found then
          - Increment N_c(SrcPhrase)
        - Else Increment TotalNumPhrases
      - If BilingualPhrase = found then
        - Increment N_f_e(BilingualPhrase)
      - Else
        - AddtoPhraseTable
        - Set N_f_e(SrcPhrase) = 1
  - For I = 1 to TotalNumPhrases
    - For every BilingualPhrase of current SrcPhrase
      - Set $P(f^* \mid \bar{e}) = \frac{N(f^*, \bar{e})}{N(\bar{e})}$
      where $N(\bar{e})$ denotes the number of times phrase $\bar{e}$ has appeared and $N(f^*, \bar{e})$ is the number of time the bilingual phrase $f^*$ & $\bar{e}$ has appeared

Figure 3.4: Phrase table generation algorithm
The basic algorithm used in the generation of the phrase translation table is shown in figure 3.4 above and we have included in Appendix B the main classes and functions used in the development of the proposed system.

<table>
<thead>
<tr>
<th>Spark plug tightening torque</th>
<th>عزم ربط شمعة الإشعال</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plug tightening torque</td>
<td>عزم ربط</td>
</tr>
<tr>
<td>Spark plug tightening</td>
<td>ربط شمعة الإشعال</td>
</tr>
<tr>
<td>tightening torque</td>
<td>عزم ربط</td>
</tr>
<tr>
<td>Spark plug</td>
<td>شمعة الإشعال</td>
</tr>
<tr>
<td>Plug tightening</td>
<td>ربط</td>
</tr>
</tbody>
</table>

**Table 3.2: Example of extracted aligned phrases**

In the figure 3.2 above there are two heuristic functions given as an input to the phrase alignment model to be used as constraint rules while extracting and calculating model parameters. These two heuristic functions were designed to get better more reliable phrases.

The first heuristic was the source phrase to target phrase length ratio. Since the word, alignment model doesn’t produce error free word alignments due to data sparseness problem. I added an upper and a lower bound to the phrase length ratio as an example it is obviously wrong to align a 4 word phrase with only one word phrase in the target and also we cannot align a 2 word phrase with 7 word long phrase in the target. This error happens usually due to data sparseness and behavior of the word alignment models in this case is to give a high fertility value for specific words and on the other hand aligns word with the NULL word. This can be shown in figure 3.5 below where we can see the word “punctuation” is aligned to four Arabic words.
The upper and the lower bound phrase length ratio is used during the phrase extraction step of the algorithm to qualify if this phrase should be added to the lexical model or not. As an example the heuristic function could be

$$|\varepsilon| : |f'| >= 1:2 \text{ and } |f'| : |\varepsilon| >= 1:2$$

| $|$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| the | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| most | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| common | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| punctuation | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| marks | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| and | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| special | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| characters | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| are | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| available | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| under | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| the | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| number | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| key | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 3.5: Wrong alignment example

The second heuristic function could be considered as adding linguistic knowledge to the phrase alignment model. It is related to the first word and the end word of the source or target phrase. As an example if we denote the first word in phrase $\varepsilon$ or phrase $f'$ as $s_1$, depending on the direction of translation we are targeting, and the last word to be $s_n$ then

$$s_1 \leftrightarrow ')' & ']' & '=' \text{ etc and } s_n \leftrightarrow '(' & '[]' & '=' \text{ etc}$$

could be considered as a constraint rule for qualifying whether to add this bilingual phrase to the phrase translation table. The main idea behind this constraint rule was to remove the noise found in the corpus.
3.3.4 Phrase Based decoder

The phrase-based decoder developed in the proposed system is different from Koehn et al [2003] which is discussed in chapter 2. The decoder developed by Koehn et al[2003] is based on the Bayes decision rule and source channel model while our decoder uses a direct approach in calculating $p(e|f)$ as follows:

$$Pr(e|f) = \arg \max \left( e^{n'} \right) \prod_{i=1}^{l} p(e'_{i} | f'_{i})$$ (1)

From the above equation we can conclude that our decoder is based on the assumption that the input sentence is segmented into a sequence of I foreign phrases. Each input phrase e'i or fi depending on the direction of translation; is translated into an equivalent output phrase. So we assume a one to one phrase alignment. The main critical differences between our decoder and the one discussed in Koehn et al[2003] is

- No phrase reordering is needed while decoding.
- No n-gram language model used.
- The decoder is using a direct approach in calculating $Pr(e|f)$

The main idea behind using the above assumptions was to use the divide and conquer approach in solving problems. If we divide the translation problem into three main sub-problems

1) Word translation choice
2) Local Word reordering within sentence segments
3) Sentence segments reordering

Then we can assume that our model will solve the first two problems by the phrase translation model discussed before and the remaining sub problem is also solved partially if the length of matching segments between the new sentence and the phrase
translation table is long enough. The third sub-problem, which is related to syntax
difference between languages, could be tackled fully in a post-processing phase
outside the scope of the decoder and this work.

These assumptions were used since we considered that the phrase based model
learns explicitly the fertility, alignment and trigram-language models and we want to
validate this in our model.

In summary, our assumption is that our phrase-based systems will learn
lexical, distortion, fertility and language models better than the word based models
due to the explicit learning of these models in our phrase translation model alone.

After this simplification to our decoder the maximization problem according
to equation (1) above has been reduced to searching for the best segmentation of the
sentence that produces the highest probability for Pr(elf). We can compose this search
problem as follows:

1) The decoder job is to extract all n-grams that appear in the test
   sentence up to a specific n according to the MaxPhraseLength used
   in the phrase translation model. As an example if the
   MaxPhraseLength used is 5 then the decoder should extract all 5, 4,
   3, 2 and unigrams found in the test sentence.

2) Then chooses the n-gram phrases that are not overlapping and
   produces the highest translation probability

3) Then for each of these n-grams the decoder should find the
   translation phrase with the highest probability from the Phrase
   translation table.

4) Finally, output the translated sentence after concatenating the
   translation phrase in the same order of the source phrases.
In figure 3.6 below you can see the search tree for this problem. In this tree, we are assuming that the MaxPhraseLength is 5. The nodes at depth 1 of the tree from left to right represents the first 5, 4, 3, 2 and 1 gram phrases respectively. The nodes at depth 2 of the tree represent the next group of n-gram phrases following the parent phrase. While you go down the tree, you are consuming the test sentence from left to right.

![Search tree](image)

**Figure 3.6: Search tree for traversing all possible n-grams in sentence**

Our decoder traverses the tree using the depth first algorithm. This algorithm was chosen since we think that the best translation will be the one that segments the sentence into fewer segments with longer phrase length. Also we wanted to generate a list of possible translation sorted based on the value of \( \text{Pr}(e|f) \).

We can see that the search space for all possible translations is huge so we designed a heuristic function in order to prune the paths that are unlikely to lead us to one of the best translations according to the number of translation options required to be generated. This heuristic function was designed in order to make the decoding process computationally tractable.
Decode(TestString, ParentNode) {
    For J = MaxPhraseLen Downto 1 {
        If (TestString > J) then { //If string length >
            Curr_Phrase = Get_Phrase(TestString, J)
        }
        //Search in Translation Table & Return highest Prob. Phrase
        GetTrans(CurrPhrase, Trans, Prob)

        Node = New Node
        Node->Length = J;
        Node->Prob = Prob
        Node->Trans = Trans

        If (ParentNode != NULL) then
            Node->AccLength = J + ParentNode->AccLength;
            Node->AccProb = Prob * ParentNode->AccProb
        Else
            Node->AccProb = Prob
            Node->AccLen = J

        //Get NewTest String after removing translated phrase
        NewString = GetNewString(TestString, CurrPhrase)
        AddChild(ParentNode, Node)

        If (SolutionsCnt <= TotalNeeded)
            || (Node->AccProb)^n >= AvgProb
        If (NewString != NULL)
            Decode(NewString, Node)
        Else
            //Back Track path to get full translation
            Translation = GetTranslation(Node)

            //Add Translation in N-bestlist Translation
            //And update AvgProb
            AddTrans(Translation, Node->AccProb)
        Else
            Do nothing (Prune Path)
    } End If
} //End For loop for children

Figure 3.7: Decoder Algorithm
The heuristic function was simply calculating the average word translation probability for the first path or the first “n” paths depending on the number of best results you want to generate then if the average word translation probability of the current path is lower than the n-best paths we backtrack and go to the next path.

The average word translation probability of the path is the $[Pr(e|f)]^{1/n}$ of the current path where $n$ is the number of translated words up to this node.

Whenever we reach the end of a path if the probability $Pr(e|f)$ is better than one of the n-best list we add this new translation and remove the one with lowest probability from the list and update the lowest average probability variable with the new probability.

In figure 3.7 we have given the algorithm of our decoder and in Appendix C of this thesis we can find the main classes and functions used in the development of this decoder.

We can say that our algorithm is a type of a Hill Climbing search where we get the first solution when traversing the tree in a depth first order then we try to find a better solution while continuing in the depth search but pruning the path that will unlikely lead us to a better solution. The advantage of our approach is that we can reach a solution very fast at the beginning of the search which also could be one of the best.
4 Experimentation

4.1 Evaluation Criteria

One of the debatable things in the machine translation community is how to evaluate system quality and results. Since a single sentence can have more than one correct translation and also since language in general also have what we call style of writing which differs from one person to another.

Since this thesis is interested in the area of computer aided translation field and how much can statistical machine translation increases productivity; I have chosen to use what is called the edit distance or the Levenshtein distance. The edit distance is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform (s) into (t). For example, if (s) is "test" and (t) is "test", then LD(s,t) = 0, because no transformations are needed. The strings are already identical. If (s) is "test" and (t) is "tent", then LD(s,t) = 1, because one substitution is needed to transform (t) to (s) (change "n" to "s"). The greater the Levenshtein distance, the more different the strings are. I have normalized the output of this algorithm with regard to string length and got the percentage of similarity instead of the difference. so as to have a more clear value to be able to do the comparison on. You can find more explanation on the edit distance and its algorithm in: http://www.merriampark.com/ld.htm

This metric is used in all the CAT tools found in the market so this will make it easier for us to know how much SMT enhanced the CAT tool suggestion to the user.
4.2 Experiment 1: IBM Model 4 versus Phrase alignment model

4.2.1 Experiment Objective

This experiment was held to see how much our phrase-based model could enhance the outcome of the translation and to analyze why this is the case and how can we increase the quality of extracted phrases.

4.2.2 Experiment Details

Within this experiment we have executed several sub experiments on our extracted corpus. As it is stated in chapter 3 for the IBM word based models we used the GIZA++ toolkit together with the CMU language modeling toolkit and the ISI rewrite greedy decoder for the training and decoding of IBM model 4. While we used our internally developed phrase extraction tool together with our phrase based decoder for the training and decoding of the phrasal translation system.

In table 4.1 we can see the number of iterations used in the training of the word based machine translation.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>2</td>
</tr>
<tr>
<td>Model 2</td>
<td>5</td>
</tr>
<tr>
<td>Model 3</td>
<td>10</td>
</tr>
<tr>
<td>Model 4</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 4.1: GIZA++ training iterations

IBM models training is based on the concept that each lower model is used to bootstrap the higher model i.e. instead of starting the training of a model with uniform parameters the parameters reached by the previous model training are used.
In all our experiments, we set the MaxPhraseLength to be 5 words. This decision was made based on the results obtained from similar experiments done [Koehn 2003] and showed that the enhancement in the quality of translation is minimal after phrase length of three.

We used the basic phrase alignment model generation tool without using any constraint regarding the source to target phrase length ratio. We just extracted the Phrase translation table based on the basic alignment file generated from the GIZA++ to be able to compare the basic phrase model with the word based models. However, in order to minimize the number of extracted phrases that will not be relevant in the translation phase we used the start word and end word constraint rule. We set this constraint to be as follows:

\[ s1 \leftrightarrow 'l' \& 'r' \text{ and } sn \leftrightarrow '(' \& 'i' \]

see section 3.3.3 for more explanation.

The test sentences used in the experiments were extracted from the corpus using a tool that is included in GIZA toolkit. This toolkit extracts 3 per 1000 sentences making sure that these test sentences are not repeated in the training set.

Using the edit distance (Levenstien distance) metric we reached the results shown in table 4.2. Appendix D includes the translation of the 38 sentences of the Mobiles corpus in the English to Arabic direction.
<table>
<thead>
<tr>
<th>Translation Direction</th>
<th>Corpus</th>
<th># of test sentences</th>
<th>Model 4 average edit distance %</th>
<th>Phrase model average edit distance %</th>
</tr>
</thead>
<tbody>
<tr>
<td>English to Arabic</td>
<td>Automotive</td>
<td>55</td>
<td>54.76</td>
<td>65.20</td>
</tr>
<tr>
<td>Arabic to English</td>
<td>Automotive</td>
<td>55</td>
<td>56.71</td>
<td>60.11</td>
</tr>
<tr>
<td>English to Arabic</td>
<td>Mobiles</td>
<td>38</td>
<td>47.20</td>
<td>77.87</td>
</tr>
<tr>
<td>Arabic to English</td>
<td>Mobiles</td>
<td>38</td>
<td>48.77</td>
<td>80.61</td>
</tr>
<tr>
<td>English to Arabic</td>
<td>Printers</td>
<td>49</td>
<td>43.50</td>
<td>47.37</td>
</tr>
<tr>
<td>Arabic to English</td>
<td>Printers</td>
<td>49</td>
<td>45.99</td>
<td>50.32</td>
</tr>
<tr>
<td>English to Arabic</td>
<td>All</td>
<td>168</td>
<td>50.58</td>
<td>60.23</td>
</tr>
<tr>
<td>Arabic to English</td>
<td>All</td>
<td>168</td>
<td>47.75</td>
<td>60.65</td>
</tr>
</tbody>
</table>

Table 4.2: Experiment 1 results

4.2.3 Results analysis

From the experiment results above it can be shown that the quality of translation of the phrase-based model is superior compared with the word based IBM models 4. The minimum enhancement was approximately 3% while the maximum was approximately 32% which is quite a large gap. We can relate this large gap to the number of long phrases matched with the test sentences (i.e. phrases of length 3 or more matching with the phrasal lexicon) see figure 4.1. The word based models didn’t make use of that since the translation is dependent on using word for word replacement.

By closely analyzing the output from both models, we can reach the following conclusions:
1. The word reordering problem is better modeled by the inclusion of this information in the phrase translation table directly. This can be shown in the following translation example taken from the output of this experiment:

   **Source:** fuel tank
   **HumanTranslation:** خزان الوقود
   **Our Translation:** خزان الوقود
   **Model 4 Translation:** الوقود خزان

   We can relate this problem to the size of the corpus used in the training.

2. Although our decoder always chooses the target phrase based only on the translation probability table calculated by the phrase translation model and ignores totally the language model effect; still our phrase decoder generates better translation syntactically and semantically than the word based decoder which uses a tri-gram language model. So we can say that our assumption that the language model is explicitly learned within the Phrase translation table is to a large extent correct except when we get a test sentence that most parts of it matches phrases of length 2 or less words. In this case also word based systems generates translation that is not useful in our goal application which is technical translation domain.
Example 1:
Source Sentence: therefore you should never rely solely upon any wireless phone for essential communications ( e.g. medical emergencies )
Target Sentence: إذا لا تحتم كمَا على الهاتف الألكسيكي في الاتصالات الضرورية ( حالات الطوارئ الطبية مثلها ).
Phrase based Translation: لا تحتم كمَا على الهاتف الألكسيكي الاتصالات الضرورية ( حالات الطوارئ الطبية مثلها ).
Model 4 Translation: أذا التي ينبغيperc طروري الضرورية لدعم يلة الألكسيكي للهاتف لمدة السلطات ( الحالات مثلها ).
Phrase based Alignment: therefore you should never rely solely upon any wireless phone for essential communications ( e.g. medical emergencies )

Example 2:
Source Sentence: if the radio is in direct mode, first activate network mode
Target Sentence: إذا كان الواديو في الوضع المباشر، فعليًا تنشيط وضع الشبكة
Phrase based Translation: إذا كان الواديو في الوضع المباشر، فعليًا تنشيط وضع الشبكة
Model 4 Translation: إذا كان الواديو في الوضع المباشر، فعليًا تنشيط وضع الشبكة
Phrase based Alignment: if the radio is in direct mode, first activate network mode

Example 3:
Source Sentence: the blinking continues for approximately 10 minutes or until the headset is connected to a compatible phone
Target Sentence: القملة استمر لمدة 10 دقائق تقريبًا، أو حتى يتم توصيل سماعة الرأس إلى هاتف متوافق
Phrase based Translation: القملة الاستمر للمدة 10 دقائق تقريبًا، أو حتى يتم توصيل سماعة الرأس إلى هاتف متوافق
Model 4 Translation: القملة استمر لمدة 10 دقائق تقريبًا، أو حتى يتم توصيل سماعة الرأس إلى هاتف متوافق
Phrase based Alignment: the blinking continues for approximately 10 minutes or until the headset is connected to a compatible phone

Figure 4.1: Translation examples from experiment 1

3. In our phrase based decoder, we didn’t use the distortion and fertility tables learned by the IBM model 4. We depended that our phrase based translation model implicitly learned that while extracting the phrases even in phrases of length 1. After analyzing the outcome from both models we found that our decoder generates better
translation than the word based decoder so this means that the fertility and distortion models in the phrase based decoding could be ignored without affecting the final outcome of translation when compared with word based decoding.

We have discovered that some of the errors in the outcome of our decoder were due to wrong word alignment coming from the word based alignment models. One of the problems with the alignment generated by word based models is due to data sparseness. For example if a rare word occurred in a sentence this sentence will end up with wrongly NULL aligned words and increasing the fertility of other words. This was shown previously in figure 3.5 where the word “punctuation” is aligned with 4 arabic words while “the”, “most” and “common” are aligned with NULL.

4.3 Experiment 2: Phrase based alignment heuristics

4.3.1 Experiment Objectives

In experiment 1 results analysis we have shown that some of the errors generated by our decoder was dependent on the wrong word alignment generated by word based alignment models. We need to experiment if by using our suggested source phrase to target phrase length ratio heuristic can we enhance the quality of translation. Also will the heuristics suggested by Och et. Al [1999] and explained in chapter 3 of this thesis enhance the quality of the proposed system output.

4.3.2 Experiment details

In this experiment, the automotive and Mobiles corpuses were used. As experiment 1 the MaxPhraseLength was 5. We have run a total of 12 experiments. For each translation direction (i.e. English to Arabic or Arabic to English) using the
mobile corpus we ran 3 experiments and the same 3 experiments was done using the automotive corpus.

<table>
<thead>
<tr>
<th>Translation Direction</th>
<th>Corpus</th>
<th>test sentences #</th>
<th>Phrase model edit distance %</th>
<th>Phrase Length ratio edit distance %</th>
<th>AndOr model edit distance %</th>
<th>AndOr length ratio edit distance %</th>
</tr>
</thead>
<tbody>
<tr>
<td>En to Ar</td>
<td>Auto</td>
<td>55</td>
<td>65.20</td>
<td>64.13</td>
<td>65.83</td>
<td>65.86</td>
</tr>
<tr>
<td>Ar to En</td>
<td>Auto</td>
<td>55</td>
<td>60.11</td>
<td>60.05</td>
<td>59.78</td>
<td>59.67</td>
</tr>
<tr>
<td>En to Ar</td>
<td>Mobiles</td>
<td>38</td>
<td>77.87</td>
<td>77.43</td>
<td>76.96</td>
<td>77.03</td>
</tr>
<tr>
<td>Ar to En</td>
<td>Mobiles</td>
<td>38</td>
<td>80.61</td>
<td>80.12</td>
<td>79.94</td>
<td>79.97</td>
</tr>
</tbody>
</table>

Table 4.3: Experiment 2 results

In the first experiment, which we named “Phrase Length ratio” in table 4.3, we used the constraint rule source phrase to target phrase length ratio parameter to be

\[
|\bar{e}| : | f^*| \geq 1:2 \text{ and } |f^*| : |\bar{e}| \geq 1:3 \tag{3}
\]

The second experiment, which we named “AndOr model” in table 4.3, uses Och heuristics in building the word alignment matrix. Finally the third experiment named “AndOr length constraint” is a combination of both i.e. we used Och heuristics in building the word alignment matrix and when generating the phrase alignment matrix we used the same constraint rule in equation (3). You can view the edit distance results of the 12 experiments compared with the basic phrase model in table 4.3.

4.3.3 Results Analysis

We found that the Andor heuristic didn’t enhance the quality of translation over the basic phrase extraction technique. However, some experiments yielded a
lower quality from the edit distance metric point of view. Also was the case with the phrase length constraint.

So we can say that these heuristics will not enhance the translation quality of the system in the application this thesis is interested in. Since the human translator will be interested in getting translation suggestions that needs the minimal addition or deletion or substitution operations (edit distance) to reach the desired correct translation.

From the previous results, we can see that these heuristics didn't affect the translation quality in a negative way to the extent that the translator can feel any difference. On the other hand, since the source to target phrase length ratio could be considered as a constraint rule that will decrease the number of possible phrases to be extracted then if we used this heuristic it will decrease the amount of memory used by our decoder. This conclusion could be seen from table 4.4 above. The results shown are for the phrases extracted from the Mobiles and the automotive corpuses once using the basic phrase translation table generation algorithm and the other after adding the source to target phrase length ratio. It is clear from the last column that the

<table>
<thead>
<tr>
<th>Translation Direction</th>
<th>Corpus</th>
<th>Phrase model</th>
<th>Phrase Length constraint</th>
<th>Difference in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>En to Ar</td>
<td>Mobiles</td>
<td>185777</td>
<td>170309</td>
<td>9.6%</td>
</tr>
<tr>
<td>Ar to En</td>
<td>Mobiles</td>
<td>199998</td>
<td>184215</td>
<td>8.1%</td>
</tr>
<tr>
<td>En to Ar</td>
<td>Auto</td>
<td>338668</td>
<td>300424</td>
<td>12.7</td>
</tr>
<tr>
<td>Ar to En</td>
<td>Auto</td>
<td>362958</td>
<td>323502</td>
<td>11.1%</td>
</tr>
</tbody>
</table>

Figure 4.4: Number of phrases extracted comparison
minimum decrease in size of the phrase translation table is 8.1% and this will lead to less memory used by the decoder.

4.4 Experiment 3: CAT tool versus SMT suggestions

4.4.1 Experiment Objectives

Since one of the main goals of this thesis is to prove that by integrating statistical machine translation system with the commercial CAT tools we can get better fuzzy match suggestions and this will lead to increasing the productivity of the human translator in the technical domain translation industry. So to prove that; an experiment was done in order to compare the fuzzy match suggested by the translation memory and the translations generated by our phrase based decoder.

4.4.2 Experiment details

The test was extracted from a new mobile phone manual. The CAT tool used in this experiment is Trados one of the industry leaders. The TM used was the same corpus our machine translation was trained on. The first 13 sentences in the manual where extracted together with the suggestions generated by the CAT tool. You will find below in table 4.5 the suggestions generated by the TM with the percentage of fuzzy match.

<table>
<thead>
<tr>
<th>Source</th>
<th>TM Suggestion</th>
<th>Fuzzy match</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Addressbook (in standby mode)</td>
<td>$\text{المؤشرات الأساسية في وضع الانتظار}$</td>
<td>44%</td>
</tr>
<tr>
<td>Display input fields</td>
<td>$\text{يحتوي أحيانا على الأقل على بيانات صالحة}$</td>
<td>0%</td>
</tr>
<tr>
<td>At least one name must be entered.</td>
<td>$\text{الرجاء إدخال رقم}$</td>
<td>38%</td>
</tr>
<tr>
<td>Always enter phone number with prefix.</td>
<td>$\text{التحقق مجموعه:}$</td>
<td>44%</td>
</tr>
<tr>
<td>Allocation to a group</td>
<td>$\text{كلمك الإطلاع على ملخص إجراء أدنا}$</td>
<td>58%</td>
</tr>
<tr>
<td>After activation, you can enter the date of birth.</td>
<td>$\text{قم بإجراء مكالمة إلى المشارك الأول.}$</td>
<td>45%</td>
</tr>
<tr>
<td>Assign a picture to the entry.</td>
<td>$\text{حفظ ملف}$</td>
<td>49%</td>
</tr>
<tr>
<td>Save the entry.</td>
<td>$\text{حذف المعلومات}$</td>
<td>38%</td>
</tr>
<tr>
<td>Additional information</td>
<td>$\text{خيارات}$</td>
<td>50%</td>
</tr>
<tr>
<td>Display options</td>
<td>$\text{إضافة إدخال}$</td>
<td>34%</td>
</tr>
<tr>
<td>Change entry</td>
<td>$\text{إدخال جديد}$</td>
<td>50%</td>
</tr>
<tr>
<td>Read entry</td>
<td>$\text{}$</td>
<td>50%</td>
</tr>
</tbody>
</table>
The number is dialled.

Table 4.5: TM fuzzy match

The same test sample was translated using our phrase based decoder to be able to compare the suggested fuzzy match translations with our translation and see how much the human translator will benefit from these translations.
Table 4.6: Edit distance between SMT & Human translation

Table 4.6 is showing the translation output of our decoder compared with the human translation for the same set of test sentences above. The last column is the edit distance between the human translation and the machine translation. In table 4.7 you can find the edit distance between the human translation and the CAT tool suggestions.

Table 4.7: TM suggestions versus human translation output
4.4.3 Results Analysis

From the experiment we can conclude that the translation outcome of our decoder was in almost all the sentences better and closer to the correct translation than what was suggested by the CAT tool. In addition, even when the CAT tool was better from the point of view of the edit distance metric it is still not usable by the translator except in one sentence. We did a survey with three experienced technical translators and we supplied them with both results we found out that translators will not make use of any suggestion that will be below 50 %. From table 4.8 you will find that according to this the translators will make use of 9 suggestions from our decoder while only one suggestion will be used from the TM.

<table>
<thead>
<tr>
<th>CAT Edit distance</th>
<th>MT Edit distance</th>
<th>Useful translations</th>
</tr>
</thead>
<tbody>
<tr>
<td>55.55</td>
<td>63.15</td>
<td>MT</td>
</tr>
<tr>
<td>0</td>
<td>23.52</td>
<td>NONE</td>
</tr>
<tr>
<td>29.54</td>
<td>20</td>
<td>NONE</td>
</tr>
<tr>
<td>21.05</td>
<td>55.26</td>
<td>MT</td>
</tr>
<tr>
<td>52.947</td>
<td>25.53</td>
<td>CAT</td>
</tr>
<tr>
<td>13.15</td>
<td>71.05</td>
<td>MT</td>
</tr>
<tr>
<td>20.51</td>
<td>65</td>
<td>MT</td>
</tr>
<tr>
<td>38.46</td>
<td>92.85</td>
<td>MT</td>
</tr>
<tr>
<td>7.142</td>
<td>100</td>
<td>MT</td>
</tr>
<tr>
<td>50</td>
<td>100</td>
<td>MT</td>
</tr>
<tr>
<td>54.54</td>
<td>100</td>
<td>MT</td>
</tr>
<tr>
<td>15.384</td>
<td>53.84</td>
<td>MT</td>
</tr>
<tr>
<td>35.89</td>
<td>31.57</td>
<td>NONE</td>
</tr>
</tbody>
</table>

Table 4.8: CAT Vs. MT Useful suggestion

From this conclusion we can suggest an architecture for an enhanced CAT environment that would increase the productivity of the translator by supplying him
Figure 4.2: SMT within CAT framework

with more accurate translation suggestions that would minimize the work needed in translating a technical document.

This architecture is shown in figure 4.3 above. After training our phrase based translation model using the corpus supplied from the TM if we have a new file supplied for translation then we first translate this file using our decoder and update the TM with the translation of the new file based on our decoder. The second stage is to supply the file to the CAT tool to be translated with a human translator. The TM will now supply the suggestion for all sentences based on the output of our decoder. If the human translator does any changes, the TM will be automatically updated with these changes. After the file is translated, we will have a new corpus which includes all new translated sentences. We can now train our model again with this new corpus and this will lead to a system that will get better over the time.
5. Conclusion

We have developed a phrase based statistical translation system. In this system we build phrase based translation model and a decoder that works with it. Our decoder simplified the translation problem based on the assumption that there is no difference in order between phrases in the target and source languages and accordingly didn't use any language model nor distortion model within the search for the translation.

We have shown that by using the translations generated by this system instead of the TM suggestions, the translator will get better suggestions and thus increase his productivity. Then we presented an architecture showing how we can include our SMT system within the framework of CAT. In this new CAT environment, we replace the TM suggestion module by our SMT system.

In addition, we have shown in our experimentations that although the corpus size was less than 20,000 sentences in certain experiments as the mobile corpus the proposed system was able to learn word reordering better than the word based models.

Moreover, we have found that by adding a simple heuristic; which limits the bilingual phrases to be extracted from the corpus based on the source and target phrase length ratio; that was suggested in this work we could decrease the size of the phrase translation table. This will make our decoder more effective from the point of view of memory usage and at the same time will not affect the quality of the translation to a noticeable percentage.

Finally, we were able to explore all the questions that was set as objectives for this thesis and stated in chapter 1. However, we still have the following topics remaining for future work. After analyzing the output and from the architecture of the proposed system it could be shown that the proposed system lacks information needed to generate syntactically accurate output. Although we have shown that our model
learns local word reordering within the sentence better than the word based SMT it remains a question how to model global word reordering within the sentence.

One idea is to add a new module as a syntax language model and experiment the proposed system with this module. This will need from us first to have an annotated Arabic corpus in order to be able to train the syntax language model then do experiments by including this module as a pre processing unit that when given a sentence it generates the same sentence but ordered according to the target language syntax. Also try the same module but after including it within the maximization problem of the decoder i.e. including it in the equation that the decoder is trying to maximize.

Other idea is to add a preprocessing module that performs morphological analysis in order to enhance the quality of the word alignment generated by word based models.

Finally, instead of using a statistical distortion and language model we can use an English-Arabic rule based syntax analyzer that will parse the English sentence and reorders the sentence according to the correct Arabic parse tree then apply our decoder on the converted sentence.

In conclusion, we want to add to the proposed system syntax knowledge about the source and target languages in order to go higher in the Vauquois triangle as shown in figure 1.1.
References List

[Al-Onaizan, 1999]

[Arturo Trujillo, 1999]

[Brown et al, 1993]

[Germann, 2001]
Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. “Fast decoding and optimal decoding for MT”. In 39th Annual Meeting of the Association for Computational Linguistics, 2001

[Hutchins, J., 1995]

[Jelinek, 2001]

[Knight, 1999a]
[Knight, 1999b]
Kevin Knight, "Decoding complexity in word-replacement translation models" Computational Linguistics, 1999

[Koehn, 2003]

[Macklovitch, 2000]

[Marcu, 2002]

[Melby, 1986]

[Nagao, 1984]

[Och, 1999]
Franz Josef Och, Christoph Tillmann, Hermann Ney. "Improved Alignment Models for Statistical Machine Translation”. In Proceeding . of the Joint
Conference. of Empirical Methods in Natural Language Processing and Very Large Corpora; University of Maryland, College Park, MD, June 1999.

[Och, 2002]

[Seasly, 2003]

[Sato et al, 1990]

[Somers, 1999]

[Venugopal, 2003]

[Weaver, 1955]
Appendix A

Part of Phrase translation table from English to Arabic alignment

| Source Phrase                                      | Target Aligned Phrase | P(f|e') | Frequency |
|----------------------------------------------------|-----------------------|--------|-----------|
| automatically starts a slide show                 | بدأ عرض الشرائح باللغة العربية | 1      | 1         |
| automatically switches on the transmitter         | تشغيل جهاز الإرسال باللغة العربية | 1      | 1         |
| automatically switching the transmitter of        | باللغة العربية | 1      | 1         |
| automatically tags starts playing the file         | بدأ قراءة الملف باللغة العربية | 1      | 1         |
| automatically update the time and                 | تحديث الوقت باللغة العربية | 1      | 1         |
| automatically with any of the                     | باللغة العربية | 1      | 1         |
| availability, performance, utilization            | تدابير توفر الأداء والتفاعل باللغة العربية | 1      | 1         |
| availability, rates and information               | تدابير توفر الأسعار والبيانات باللغة العربية | 1      | 1         |
| availability and a subscription to                | حصول على معلومات حول إمكانية الاشتراك باللغة العربية | 1      | 1         |
| availability and the synchronisation service      | توفر خدمة التنسيق وضبطها باللغة العربية | 1      | 1         |
| availability may vary by country                  | توفر رمضان باللغة العربية | 1      | 1         |
| availability of an operator logo                  | توفر شعار الشبكة باللغة العربية | 1      | 1         |
| availability of approved accessories               | حصول على قائمة بالمستلزمات المعتمدة باللغة العربية | 1      | 1         |
| availability of approved batteries and            | حصول على بطاريات وأجهزة باللغة العربية | 1      | 1         |
| availability of approved chargers                 | حصول على أجهزة شحن معتمدة باللغة العربية | 1      | 1         |
| availability of different wap services            | توفر خدمات مختلف النطاق باللغة العربية | 0.5    | 1         |
| availability of particular products may           | توفر منتجات معينة باللغة العربية | 1      | 1         |
| availability of the settings                       | توفر هذا الضيف باللغة العربية | 1      | 1         |
| availability of wap services                       | wap | 0.5    | 1         |
| available for a game or                           | المتاحة للعبة باللغة العربية | 1      | 1         |
| available for an application or                   | متاحة لتطبيق باللغة العربية | 2      | 1         |
| available for control signaling in                | متاحة للتحكم باللغة العربية | 1      | 1         |
| available for control signaling when              | المتاحة للتحكم باللغة العربية | 1      | 1         |
| available for different phone models              | المتاحة لطرازات الهواتف المتعددة باللغة العربية | 1      | 1         |
| available for download from the                   | المتاحة للتحميل باللغة العربية | 1      | 1         |
| available for the divert option                   | متاحة في الخروج باللغة العربية | 1      | 1         |
| available for the languages in                    | المتاحة للغات باللغة العربية | 2      | 1         |
| available for the phone                           | المتاحة للهاتف باللغة العربية | 1      | 1         |
| available for your phone                          | المتاحة للهاتف باللغة العربية | 1      | 1         |
| available functions is shown                      | المتاحة للهاتف باللغة العربية | 1      | 1         |
| available in http : /                            | http : / | 1      | 1         |

65
<table>
<thead>
<tr>
<th>averaged over ten grams of</th>
<th>بالمعتل لشرة</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>avoid potential interference with the</td>
<td>وتجنب التداخل مع البيئة</td>
<td>1</td>
</tr>
<tr>
<td>avoid similar names for different</td>
<td>وتجنب استخدام الأسماء المتشابهة لأرقام مختلفة</td>
<td>1</td>
</tr>
<tr>
<td>avoid the ned to queue</td>
<td>وتجنب الانتظار حتى يتم</td>
<td>0.5</td>
</tr>
<tr>
<td>avoid the ned to queue</td>
<td>وتجنب الحاجة إلى الانتظار حتى يتم</td>
<td>0.5</td>
</tr>
<tr>
<td>avoid using acces codes similar</td>
<td>تتبع استخدام رموز الوصول المتشابهة لأرقام</td>
<td>1</td>
</tr>
<tr>
<td>away from smal children</td>
<td>بعيدا عن الأطفال الصغار</td>
<td>1</td>
</tr>
<tr>
<td>away from the camera to</td>
<td>بعيدا عن الكاميرا</td>
<td>1</td>
</tr>
<tr>
<td>away from the face and</td>
<td>بعيدا عن الوجه</td>
<td>1</td>
</tr>
<tr>
<td>away from the mouth with</td>
<td>من الفم بعيدا بحيث يتجه</td>
<td>1</td>
</tr>
<tr>
<td>away from your face and</td>
<td>بعيدا عن الوجه</td>
<td>1</td>
</tr>
<tr>
<td>away from your mouth with</td>
<td>من الفم بعيدا بحيث يتجه</td>
<td>1</td>
</tr>
<tr>
<td>b , or c )</td>
<td>b أو c</td>
<td>1</td>
</tr>
<tr>
<td>b . ad text :</td>
<td>ب . إضافة (نص ) :</td>
<td>0.5</td>
</tr>
<tr>
<td>b . ad text :</td>
<td>ب . إضافة (نص ) :</td>
<td>0.5</td>
</tr>
<tr>
<td>b . click on the</td>
<td>ب . انقر فوق</td>
<td>1</td>
</tr>
<tr>
<td>b . to select part</td>
<td>ب . للتحديد جزء</td>
<td>1</td>
</tr>
<tr>
<td>b and c ) on</td>
<td>b و c</td>
<td>1</td>
</tr>
<tr>
<td>b or c ) of</td>
<td>b أو c</td>
<td>1</td>
</tr>
<tr>
<td>back - up data from</td>
<td>احتياطية منها</td>
<td>1</td>
</tr>
<tr>
<td>back #/ dt # .</td>
<td>&quot; &quot;</td>
<td>1</td>
</tr>
<tr>
<td>back #/ dt # and</td>
<td>&quot; &quot;</td>
<td>1</td>
</tr>
<tr>
<td>back #/ dt # or</td>
<td>&quot; &quot;</td>
<td>1</td>
</tr>
<tr>
<td>back #/ dt # to</td>
<td>&quot; &quot;</td>
<td>1</td>
</tr>
<tr>
<td>back and downwards so that</td>
<td>باتجاه أسفل</td>
<td>1</td>
</tr>
<tr>
<td>back and pres ( graphic</td>
<td>اليد على #</td>
<td>1</td>
</tr>
<tr>
<td>back cover of the phone</td>
<td>غطاء الهاتف الخلفي</td>
<td>0.25</td>
</tr>
<tr>
<td>back cover of the phone</td>
<td>غطاء الهاتف الخلفي</td>
<td>0.5</td>
</tr>
<tr>
<td>back cover of the phone</td>
<td>غطاء الهاتف الخلفي بعيدا عن الهاتف</td>
<td>0.25</td>
</tr>
<tr>
<td>back cover so that the</td>
<td>غطاء الهاتف الخلفي</td>
<td>1</td>
</tr>
<tr>
<td>back from privacy mode to</td>
<td>من وضع الخصوصية إلى</td>
<td>1</td>
</tr>
<tr>
<td>back key in direct mode</td>
<td>الحاجة في الوضع المبكر</td>
<td>1</td>
</tr>
<tr>
<td>back key in the middle</td>
<td>الحاجة الموجودة في وسط</td>
<td>1</td>
</tr>
<tr>
<td>back key selects the first</td>
<td>الحاجة إلى تحديد أول</td>
<td>1</td>
</tr>
<tr>
<td>back key switches between the</td>
<td>ذات الحاجة إلى الانتقال بين</td>
<td>0.25</td>
</tr>
<tr>
<td>back key switches between the</td>
<td>الحاجة إلى الانتقال بين</td>
<td>0.25</td>
</tr>
<tr>
<td>back key switches between the</td>
<td>الحاجة إلى الانتقال بين</td>
<td>0.5</td>
</tr>
<tr>
<td>back of the nokia image</td>
<td>الهاتف</td>
<td>1</td>
</tr>
<tr>
<td>back of the phone ,</td>
<td>الهاتف</td>
<td>1</td>
</tr>
<tr>
<td>back of your computer and</td>
<td>هاتف الكمبيوتر</td>
<td>1</td>
</tr>
<tr>
<td>back of your pc .</td>
<td>هاتف الكمبيوتر</td>
<td>1</td>
</tr>
<tr>
<td>back on by pressing any</td>
<td>الضغط على أي</td>
<td>1</td>
</tr>
<tr>
<td>back the settings to the</td>
<td>الإعدادات إلى</td>
<td>1</td>
</tr>
<tr>
<td>back to arabic text input</td>
<td>لعدة إلى وضع إدخال اللغة العربية</td>
<td>1</td>
</tr>
<tr>
<td>back to go to the</td>
<td>الانتقال إلى</td>
<td>1</td>
</tr>
<tr>
<td>back to return to the</td>
<td>الانتقال إلى</td>
<td>1</td>
</tr>
<tr>
<td>back to the homepage of</td>
<td>الصفحة الرئيسية الخاصة</td>
<td>1</td>
</tr>
<tr>
<td>back to the previous ones</td>
<td>المتغير السابق</td>
<td>1</td>
</tr>
<tr>
<td>back to the previous page</td>
<td>إلى الصفحة السابقة</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
<td></td>
</tr>
<tr>
<td>back to the support pages</td>
<td>إلى صفحات الدعم</td>
<td></td>
</tr>
<tr>
<td>back up data on your</td>
<td>حفظ نسخة احتياطية من البيانات الموجودة</td>
<td></td>
</tr>
<tr>
<td>back up information on a</td>
<td>عملي تنسيخ احتياطية من المعلومات المسجلة</td>
<td></td>
</tr>
<tr>
<td>back up information on a</td>
<td>عملي تنسيخ احتياطية من المعلومات المسجلة</td>
<td></td>
</tr>
<tr>
<td>background and a non -</td>
<td>بينما تقلل</td>
<td></td>
</tr>
<tr>
<td>background folder #/ dt #</td>
<td>حافظة الاختيار</td>
<td></td>
</tr>
<tr>
<td>background groups are used for</td>
<td>استخدم المجموعات</td>
<td></td>
</tr>
<tr>
<td>background groups in scanning</td>
<td>والمجموعات الخفية ضمن عملية المصد</td>
<td></td>
</tr>
<tr>
<td>background image , known as</td>
<td>صورة خفية تعرف</td>
<td></td>
</tr>
<tr>
<td>background images in the phone</td>
<td>صورة خفية في الهاتف</td>
<td></td>
</tr>
<tr>
<td>background lights changes randomly</td>
<td>الأضواء الخفية بطريقة عشوائية</td>
<td></td>
</tr>
<tr>
<td>background picture , wallpaper ,</td>
<td>صورة خفية أو ورق حافظة</td>
<td></td>
</tr>
<tr>
<td>background when performing other operations</td>
<td>الخفية أثناء أداء عمليات أخرى</td>
<td></td>
</tr>
<tr>
<td>backslash \ ) acros</td>
<td>شرطة ماتئة (1) على</td>
<td></td>
</tr>
<tr>
<td>backslash \ ) acros</td>
<td>وضع شرطة ماتئة (1) على</td>
<td></td>
</tr>
<tr>
<td>backup copies of all important</td>
<td>عملي تنسيخ احتياطية من جميع</td>
<td></td>
</tr>
<tr>
<td>backup copies of all important</td>
<td>تنسيخ احتياطية من جميع</td>
<td></td>
</tr>
<tr>
<td>backup copies of your images</td>
<td>عملي تنسيخ احتياطية من الصور الخاصة</td>
<td></td>
</tr>
<tr>
<td>backwards ( 4 ) and</td>
<td>خلف (4)</td>
<td></td>
</tr>
<tr>
<td>backwards , pres ( graphic</td>
<td>نضغط على (</td>
<td></td>
</tr>
<tr>
<td>backwards and down or up</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bag , remember that an</td>
<td>الوفقية ، تذكر أن</td>
<td></td>
</tr>
<tr>
<td>bag inflates , serious injury</td>
<td>السرعة ، ربما ينتج عن ذلك إصابة خطيرة</td>
<td></td>
</tr>
<tr>
<td>bag inflates with great force</td>
<td>السرعة تتغلق قوة مهيبة</td>
<td></td>
</tr>
<tr>
<td>balance settings to get the</td>
<td>موزونة الوضع لوصول إلى</td>
<td></td>
</tr>
<tr>
<td>bank account ) , empty</td>
<td>قفل بتفريغ</td>
<td></td>
</tr>
<tr>
<td>banking , news , weather</td>
<td>الخدمات المصغرة والأخبار</td>
<td></td>
</tr>
<tr>
<td>banking services , and for</td>
<td>الخدمات المصغرة</td>
<td></td>
</tr>
<tr>
<td>banking services , you ned</td>
<td>الخدمات البنكية التي تحتاج</td>
<td></td>
</tr>
<tr>
<td>banking services or shopping on</td>
<td>الخدمات المصغرة أو التسوق من خلال أحد</td>
<td></td>
</tr>
<tr>
<td>bar ( 1 ) .</td>
<td>(1)</td>
<td></td>
</tr>
<tr>
<td>bar ( 1 ) and</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>bar , the more power</td>
<td>للذكاء على كثرة الطاقة</td>
<td></td>
</tr>
<tr>
<td>bar , the stronger the</td>
<td>للذكاء على قوة</td>
<td></td>
</tr>
<tr>
<td>bar above ( graphic )</td>
<td>الشريط الموجود أعلى (</td>
<td></td>
</tr>
<tr>
<td>bar is shown on the</td>
<td>عرض الشريط على</td>
<td></td>
</tr>
<tr>
<td>baring password ( 4 digits</td>
<td>كلمة سر الحظر ( 4 أرقام</td>
<td></td>
</tr>
<tr>
<td>baring password is neded when</td>
<td>يلزم إدخال كلمة سر الحظر عند</td>
<td></td>
</tr>
<tr>
<td>baring password is required</td>
<td>يلزم إدخال كلمة سر الحظر</td>
<td></td>
</tr>
<tr>
<td>baring service #/ dt #</td>
<td>سر الحظر</td>
<td></td>
</tr>
<tr>
<td>baring service #/ dt #,</td>
<td>سر الحظر</td>
<td></td>
</tr>
<tr>
<td>bars of nokia pc suite</td>
<td>nokia pc suite</td>
<td></td>
</tr>
<tr>
<td>base station ( bs )</td>
<td>محطة الأساسية (</td>
<td></td>
</tr>
<tr>
<td>base station ( radio unit</td>
<td>محطة راديوية (</td>
<td></td>
</tr>
<tr>
<td>base station ( s )</td>
<td>المحطات الرئيسية</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

Phrase Alignment Generation Source Code
Main Classes Definition

//----------------------------------------------------------------------------------------
// Main Classes
#ifndef LexH
#define LexH
#include <StrUtils.hpp>
#include <sysdyn.h>
#include <stdio.h>
#include <io.h>
#include <alloc.h>
#include <fcntl.h>
#include <process.h>
#include <sys/stat.h>
#include <Classes.hpp>
#include <StdCtrls.hpp>
#include "Matrix.h"
//----------------------------------------------------------------------------------------
/*
TLex class is responsible for analyzing single matrix
and getting all valid Bilingual Phrases of specific Phrase Length
*/

class TLex
{
private:
    int LexLen;//Phrase Length
    int MinimumRation;
    int MaximumRation;
    String StdDelimt;
    String EndDelimt;
    TMatrix *Matrix;//Matrix object to be converted to lexicon item
    TStringList *SrcDelimt;//List of Src words index
    TStringList *TargetDelimt;//List of trg words index
    TStringList *SemiLexlt;//Memory Phrase table
    TStringList *StdDelimtlst;
    TStringList *EndDelimtlst;
    int targetFirst;
    int MaxIndex,MinIndex;// min. and max. numbers
        // used to get the NULL aligned words
//function used to add leading zeros to an integer e.g. 1 become "001"

String str(int i);

//function used to check if valid phrase to add it to Mem. Phrase Table
int ValidPhrase();

// returns a string before a specific delimiter
String GetDelimit(String* StrSource,String Delimit);

public:
    TLex(int Length,TMatrix *Mx,TStringList *SLexlst,int MinRation,
         String SD,String ED,int Tgtfirst,int MaxRation);
    ~TLex();

    void GetLexItems();
    void GetLexItem(int Index);
    void GetLexItemsInBetween();
    void WriteItem(); //writes items to Mem. Phrase table
    void Fill_delimit_list();

};
//@--------------------------------------------------------
#endif

Main Classes Implementation
//@--------------------------------------------------------

#pragma hdrstop
#include "Lex.h"

//@--------------------------------------------------------
TLex::TLex(int Length, //SrcPhraseLen
            TMatrix *Mx, //Word aligned Matrix
            TStringList *SLexlst, //Memory Phrase Table
            int MinRation, // Min SrcToTrgt phrase Ratio
            String SD,String ED, //Start And end Delimiters
            int Tgtfirst, //Not used
            int MaxRation) // Min SrcToTrgt phrase Ratio
{
    LexLen=Length;
    Matrix=Mx;
    TargetList=new TStringList;
    SourceList=new TStringList;
    SemiLexList= SLexlst;
    MinimumRation=MinRation;
    MaximumRation=MaxRation;
Stdelimt=SD+"\",);
Enddelimt=ED+"\",);
Stdelimtlist=new TStringList;
Enddelimitlist=new TStringList;
Fill_delimit_list();
targetFirst=Tgtfirst;
}
//-------------------------------------------------------------------------------
TLex::~TLex()
{
    delete SourceList;
    delete TargetList;
    delete Stdelimitlist;
    delete Enddelimitlist;
}
//-------------------------------------------------------------------------------
void TLex::GetLexItems()
{
    // a loop to get all source segments combination to be added to phrase
    // table
    for (int i=0;i<Matrix->xlen-LexLen+1;i++)
    {
        // clear lists
        SourceList->Clear();
        TargetList->Clear();
        // init max and min indexes
        MaxIndex=-1;
        MinIndex=-1;
        // a loop to add source words and its aligned targets
        for (int j=0;j<LexLen ;j++)
        {
            // add source word
            SourceList->Add(str(i+j));
            // call function to get aligned words
            GetLexItem(i+j);
        }
        // check if there is null aligned words in between
        if(MinIndex>=-1)
            GetLexItemsInBetween();
        // check if target phrase is cont.
        if (ValidPhrase())
            // add item to memory phrase table
            WriteItem();
    }
}

//-------------------------------------------------------------------------------

70
// get aligned target words for a specific source word

void TLex::GetLexItem(int Index) {//index of source word

  
  for (int j=0;j<Matrix->ylen;j++)
  
    if (Matrix->Data[Index][j]="1")
    
      if (TargetList->IndexOf(str(j))==1)
      
        //add target word index to target list
        TargetList->Add(str(j));
        //reset min and max indexes
        if (j>MaxIndex)
          MaxIndex=j;
        if (j<MinIndex||MinIndex==-1)
          MinIndex=j;
    
  
}


//get all nul aligned words between min and max index
void TLex::GetLexItemsInBetween()

  
  for (int i=MinIndex+1;i<MaxIndex;i++)
  
    int Assigned=0;
    for (int j=0;j<Matrix->xlen;j++)
      
        if (Matrix->Data[j][i]="1")
          Assigned=1;
 
        if (Assigned==0)
          TargetList->Add(str(i));
  
}

//write source and target segments to phrase table memory
void TLex::WriteItem()

  
  SourceList->Sort();
  TargetList->Sort();
  String Line;
  String srcLine=Matrix->Source[SourceList->Strings[0].ToInt()];
  for (int i=1;i<(SourceList->Count);i++)
  

srcline=""+Matrix->Source[Sourcelst->Strings[i].ToInt()];
}

String tgtline;
if (Targetlst->Count >0)
  tgtline=Matrix->Target[Targetlst->Strings[0].ToInt()];

for (int j=1;j<(Targetlst->Count);j++)
{
  tgtline=""+Matrix->Target[Targetlst->Strings[j].ToInt()];
}
if (targetFirst==1)
  Line= tgtline+"\t"+srcline;
else
  Line= srcline+"\t"+tgtline;

SemiLexlst->Add(Line);

//--- check if phrase is valid
int TLex::ValidPhrase()
{
  int res=1;
  // if it has no translation then it is not valid
  if (Targetlst->Count==0)
    return 0;
  // if target to source ratio length is less than min ratio then it is
  // not valid
  if ((1.00*Targetlst->Count/Sourcelst->Count)<(1.00*MinimumRation/100))
    return 0;
  // if target to source ratio length exceeded max ratio
  // then it is not valid
  if ((1.00*Targetlst->Count/Sourcelst->Count)>(1.00*MaximumRation/100))
    return 0;
  // if segment starts with a linguisticsall not correct word
  // then it is not valid
  if (Stdelimlst->IndexOf
    ((Matrix->Source [Sourcelst->Strings[0].ToInt()])[1 ])!=-1)
    return 0;
  // if segment ends with a linguisticsall not correct word
  // then it is not valid
  if (Enddelimlst->IndexOf((Matrix->Source[Sourcelst->Strings
    [Sourcelst->Count-1].ToInt()])[1])!=-1)
    return 0;
  //check on continuity
  for (int i=MinIndex+1;i<MaxIndex;i++)
  {
    if (Targetlst->IndexOf(str(i))==1)
  
72
res=0;
}
return res;

}
//...............................................................................
String TLex::str(int i)
{
    String res;
    if (i >99)
        res=IntToStr(i);
    else if (i >9)
        res="0"+IntToStr(i);
    else
        res="00"+IntToStr(i);
    return res;
}
//...............................................................................

void TLex::Fill_delimit_list()
{
    while (Stdelimit!="")
    {
        Stdelimitlst->Add(GetDelimit(&Stdelimit,"",""));
    }
    while (Enddelimit!="")
    {
        Enddelimitlst->Add(GetDelimit(&Enddelimit,"",""));
    }
}
//...............................................................................
String TLex::GetDelimit(String* StrSource, String Delimit)
{
    String result=MidStr(*StrSource,0, StrSource->AnsiPos(Delimit)-1);
    *StrSource=MidStr(*StrSource,
        StrSource->AnsiPos(Delimit)+Delimit.Length(),StrSource->Length());
    return (result);
}
//...............................................................................

void TMatrices::GetMatrices_to_Lex(
    String SLex_file,
    String Lex_file,
    int Sourcecnt,
    int MinRation,
    String SD,String ED,
    int Tgtfirst,int MaxRation)
{ TStringList *LexlstPart=new TStringList; //a temp. phrase table
    //to be filled with one matrix
    //bilingual phrases

TStringList *SLexlst=new TStringList; // memory phrase table

TStringList *Lexlst=new TStringList; // Phrase table

int MatrixNum=Readfromlist(Datalines);
for (int i = 0 to MatrixNum)
{
    TLex *Lex=new TLex(Sourcecnt,Matrix[i],LexlstPart,
    MinRation,SD,ED,Tgtfirst,MaxRation);
    Lex->GetLexItems();
    delete Lex;
    //add current temp memory phrase table to full memory phrase
    SLexlst->AddStrings(LexlstPart);

    LexlstPart->Clear(); //clear temp for new sentence
}

SLexlst->Sort(); //Sort Memory Phrase Table to get probabilities
Compute_Phrase_prob(SLexlst,Lexlst);
Lexlst->SaveToFile(Lex_file);
delete SLexlst;
delete Lexlst;

}

//-----------------------------------------------

#pragma package(smart_init)
Appendix C

Decoder Main Source Code

Main Classes Definition

//-----------------------------------------------

#ifndef testH
#define testH
#include <StrUtils.hpp>
#include <sysdyn.h>
#include <stdio.h>
#include <cio.h>
#include <alloc.h>
#include <fcntl.h>
#include <process.h>
#include <sys\stat.h>
#include <Classes.hpp>
#include <StdCtrls.hpp>
#include <StrUtils.hpp>
#include "Lex.h"
#include <ComCtrls.hpp>

// NodeDataStruct is A Structure to store every segment node data in // the search tree
typedef struct NodeDataStruct
{
    String Translation;    //stores segment translation
    float Probability;     //stores segment probability
    float AcumProbability; //stores Accumulated segments probability in the path
    int Length;            //word length of current segment
    int AcumLength;        //word length of current path
} NodeData;
typedef NodeData* PnodeData;

/*-----------------------------------------------
this class gets test sentences from file and create a
ittestitem object for each sentence to get translation
-----------------------------------------------------------------*/
class TTest
{
private:
public:
TStringList *TestLines; //test sentences from file.
TTest(String filename); //filenem= name of test file
~TTest();

void GetTestItems(TLexLst *Lexlst[],int MaxPhraseLength, int TotNumSol);

};

/**********************************************************************

TTestItem

Tests items class process test string and get translations
for this String

**********************************************************************/

class TTestItem
{
private:

TStringList *Translations;//Array that saves all n-best translations
int MaxPhraseLen; // the max. phrase length in Phrase Table objects
int SolutionsCnt; //the found translations count
int TotalNeededSolutions; //maximum need translation count.
int CompletedPaths;//number of completed paths
int UnCompletedPaths;//number of incomplete paths
double AvgProbability;//the current average word probability.

//a suplementary function that searches for a delimeter in a string
//and returns all Preceeding chars
String GetDelimit(String* StrSource,String Delimit);

//backtrack path to get complete translation
//to save it in the translation table .
Get_Total_Trans(TTreeNode *Node);

//Add term in the translation table ordered by probability.
Add_Term(String Translation,float Probability);

Get_Word_List(TStringList *Word_List,String SourceString);
String Get_Segment(TStringList *Word_List,int WordCount)

public:

TTestItem(int PTotalitno, int PMaxPhraseLen);

// Search algorithm generates translation tree.
int GetSegmentTrans(TLexLst *Lexlst[],String TestRemainigString,
TTreeNode *ParentTransNode);
~TTestItem();

};
TTreeView *TransSegTree;

#include "test.h"
#include "Unit4.h"
#include "Math.h"

#pragma hdrstop

// Load test sentences from file
TTest::TTTest(String filename)
{
    TransSegTree= new TTreeView;
    TestLines= new TStringList;
    TestLines->LoadFromFile(filename);
}

TTest::~TTTest()
{
    delete TestLines;
    delete TransSegTree;
}

// Function loops on testlines and generate testitem object
// for each to get translations

void TTest::GetTestItems(
    TLexLst *LexLst[], //array of Phrase table objects
    int MaxPhraseLength, //maximum phrase length that could be
                        //translated from Phrase table objects.
    int TotNumSol //number of needed translation option per sent.
)
```cpp
{  
    String CurStr;
    for (int i=0;i<TestLines->Count;i++)  
    {
        //Create New Test sentence Object
        TTestItem *TestItem=new TTestItem(TotNumSol, MaxPhraseLength);

        // Create New node to be the root node for this test sentence
        TTreeNode *ChNode=
            TransSegTree->Items->AddChild(TransSegTree->Items->Item[0],
            TestLines->Strings[i]);

        //Start Translation search
        TestItem->GetSegmentTrans(LexLst, TestLines->Strings[i], ChNode);
    }
}

//----------------------------------------------------------------------------------
TTestItem::TTestItem(int PTotalitno, int PMaxPhraseLen)
{
    Translations=new TStringList;
    AvgProbability=0;
    CompletedPaths=0;
    UnCompletedPaths=0;
    SolutionsCnt=0;
    TotalNeededSolutions= PTotalitno;
    MaxPhraseLen = PMaxPhraseLen;
}
//----------------------------------------------------------------------------------
TTestItem::~TTestItem()
{
    delete Translations;
}

// This is the implementation of the decoder search algorithm
//
int TTestItem::GetSegmentTrans(
    TLexLst *LexLst[], //pointer to array of Phrase tables Obj.
    String TestRemainingString, //Remaining string
    TTreeNode *ParentTransNode) //Parent node
{
    TStringList *TestRemainingWords=new TStringList; // array of remaining words
    String Translation;
    ```
float Probability;
String CurrSegment;
String NewRemainingStr;

// add remaining words to remaining array.
Get_Word_List(TestRemainingWords, TestRemainingString);

// Loop for all possible segments of length 1 to MaxPhraseLen from the
//begining of Remaining part of TestSentence
for (int j=0; j<MaxPhraseLen; j++)
{
    // Check if Len of Remaining words >= Current PhraseLen
    if (TestRemainingWords->Count > MaxPhraseLen-j)
    {
        Translation="";
        CurrSegment=Get_Segment(TestRemainingWords, MaxPhraseLen-j);

        // searches for the segment in the Phrase table and
        // return translation and probability
        Probability=LexList[j]->GetTrans(CurrSegment,&Translation);

        // a new node to save current segment translation data
        TTreeNode *ChNode=
            TransSegTree->Items->AddChild(ParentTransNode, CurrSegment);

        // saving data in the node
        NodeData *Ndata= new NodeData;
        Ndata->Translation=Translation;
        Ndata->Probability=Probability;
        Ndata->Length=MaxPhraseLen-j;

        // If Parent is Not Root Node
        if (ParentTransNode->Data!=0)
        {
            Ndata->AcumLength=Ndata->Length +
                Pnodedata(ParentTransNode->Data)->AcumLength;

            Ndata->AcumProbability=Probability *
                (Pnodedata(ParentTransNode->Data)->AcumProbability);
        }
        else
        {
            Ndata->AcumProbability=Probability;
            Ndata->AcumLength=Ndata->Length;
        }
        ChNode->Data=Ndata;
}
// get remaining string to be passed to the same function to
// go one level down in the tree
   NewRemainingStr=TestRemainingString;
   GetDelimit(&NewRemainingStr,CurrSegment);

// check if the list of best trans. is not yet full OR
// the current word probability is greater than history avg.
if ((SolutionsCnt<=TotalNeededSolutions)
   ||(pow(Ndata->AcumProbability, 1.0/Ndata->AcumLength)
      >=AvgProbability))
{
   if (Trim(NewRemainingStr)!="")
   {
      GetSegmentTrans(LexIst,NewRemainingStr,ChNode);
   }
   else // last node
   {
      //get total translation & Add Trans to N-Best
      //list
      Get_Total_Trans(ChNode);
      CompletedPaths++;
   }
   else // Prune this path
   UnCompletedPaths++;

}(Yii (TestRemainingWords->Count > MaxPhraseLen-j)
} //for (int j=0; j<MaxPhraseLen; j++)
delete TestRemainingWords;
}

//------------------------------------------------------------------------
// String TTestItem::GetDelimit(String* StrSource, String Delimit)
{
   String result=MidStr(*StrSource,0,StrSource->AnsiPos(Delimit)-1);
   *StrSource=MidStr(*StrSource,StrSource->AnsiPos(Delimit)+ Delimit.Length(),
                StrSource->Length());
   return (result);
}

//------------------------------------------------------------------------

TTestItem::Get_Total_Trans(TTreeNode *Node)
{
   TTreeNode *PNode=Node;
   
}
String TotalTranslation=PNodedata(PNode->Data)->Translation;
// backtrack segment translations for this path to get full translation
while (PNode->Parent->Data!=0)
{
    PNode=PNode->Parent;
    TotalTranslation=PNodedata(PNode->Data)->Translation+" "+TotalTranslation;
}
TotalTranslation=TotalTranslation
    + "\t" +
    PNodedata(Node->Data)->AcumProbability
    + "\t" +
   FloatToStr(pow(PNodedata(Node->Data)->AcumProbability,
                  1.0/PNodedata(Node->Data)->AcumLength));

// Add translation in ordered list of n-best translation
Add_Term(TotalTranslation,PNodedata(Node->Data)->AcumProbability);
    SolutionsCnt++;
}
//---------------------------------------------------------------------------
TTTestItem::Add_Term(String Translation,float Probability)
{
    String CurrProb;
    String CurrString;
    int GreaterFound=0;
    // add new entry in translation list with probability order
    if (Translations->Count==0)//first entry
    {
        Translations->Add(Translation);
    }
    else
    {
        // loop to get the right position (in probability order)
        // to add the new translation
        for (int i=0;i<Translations->Count ;i++)
        {
            CurrString=Translations->Strings[i] ;
            GetDelimit(&CurrString, "\t");
            CurrProb=GetDelimit(&CurrString, "\t");
            if (Probability> StrToFloat(CurrProb))
            {
                GreaterFound =1;
                Translations->Insert(i,Translation);
                i=Translations->Count; //break;
            }
        }
    
    // if it is the least probability so i add it as the last one
    if (GreaterFound ==0)
```cpp
{
    Translations->Add(Translation);
}

// check if the count exceeded total count needed
// so delete the last one (least probability)
if (Translations->Count>TotalNeededcSolutions)
{
    Translations->Delete(TotalNeededcSolutions);
}

// Get New Avg word probability
CurrString=Translations->Strings[Translations->Count-1];
GetDelimit(&CurrString,"\t");
GetDelimit(&CurrString,"\t");
AvgProbability=StrToFloat(CurrString); // save new probability

//-------------------------------------------------------------------------------

TTestItem::Get_Word_List(TStringList *Word_List, String SourceString);
{
    String Temp=SourceString+" ";
    while (Temp !="")
        Word_List->Add(GetDelimit(&Temp, " "));
}

//-------------------------------------------------------------------------------

String TTestItem::Get_Segment(TStringList *Word_List, int WordCount);
{
    String Segment=Word_List->Strings[0];
    for (int z=1; z<WordCount; z++)
    {
        Segment=Segment+" "+TestRemainingWords->Strings[z+1];
    }
    return (Segment);
}

//-------------------------------------------------------------------------------

#pragma package(smart_init)
```
**Appendix D**

**Sample Translations**

**Translation Direction: English to Arabic**
**Corpus: Mobile**
**Model Used: IBM model 4**

<table>
<thead>
<tr>
<th>Human Translation</th>
<th>Model 4 Translation</th>
<th>Edit Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>أتائنا استخدام مصيدة أسماة الأبال أصل الرويدد كاي</td>
<td>عند الآن استخدم الإستراحه الرودرى كما التى فعله أي</td>
<td>43.40</td>
</tr>
<tr>
<td>هاتف آخر مع معاينة أن الهوية منتهى إلى الالتحى</td>
<td>مع الهوية منتهى يصل مع مصور أب يل الكلب</td>
<td></td>
</tr>
<tr>
<td>هذا الرودرى كان هاتف للإلكترى، وهم يستخدم إبرادات الدوهي</td>
<td>هذا الرودرى، تعتبر أي إلكترونية الهاتف، تتم استخدام الراديو إشارة الإلكترونية مع الأراضى التسويكات كما أن هناك المستخدم، والبطون وطابع</td>
<td>41.01</td>
</tr>
<tr>
<td>إذا لا تتعابك على الهوائي أو إلكترى في الاتصالات الضوئية (حالات الطوارئ الطبية)</td>
<td>لمدة مطلوبة للإتصالات (مثل الجملة</td>
<td>41.44</td>
</tr>
<tr>
<td>هذا السباق كمكات الطوارئ أو مكبات على كافية، شاشة الهواتف الإلكترونية أو ما تكون مع بعض حقن الشبكة و/أو أقسم بالوردى قبل الاستخدام</td>
<td>الطوارئ كمكات قد لا تكون قبل قدر جميع الإلكترونية الهاتف، الشاشة أو ما تكون الشبكة تحدد مع و/أو الوردى مزأر يتم في استعمال</td>
<td></td>
</tr>
<tr>
<td>تأكد من ذلك في المرآة المحمية للخدمة الخلوية</td>
<td>تفاصيل مع المحمية الخدمة موزود</td>
<td>43.48</td>
</tr>
<tr>
<td>يمكن إجراء كميات الطوارئ فقط في وضع</td>
<td>الطوارئ كميات يمكن فقط كجزء من الشبكة وضع</td>
<td>38.89</td>
</tr>
<tr>
<td>إذا كان الوردى في وضع المبادئ، فم لا</td>
<td>إذا كان الوردى في وضع المبادئ، فم لا تتشيب وضع الشبكة</td>
<td>39.29</td>
</tr>
<tr>
<td>إذاً إخراج كميات الطوارئ:</td>
<td>إذاً إخراج كميات الطوارئ:</td>
<td>56.92</td>
</tr>
<tr>
<td>إذا الواردى لا فص في وضع المبادئ</td>
<td>إذا الواردى لا فص في وضع المبادئ</td>
<td>45.45</td>
</tr>
<tr>
<td>تأكد من أن الشبكة ممدوة.</td>
<td>تأكد من أن الشبكة ممدوة.</td>
<td>43.24</td>
</tr>
<tr>
<td>استخدم على (مثل graphic) عدد مرات (مثل graphic) كما العديده ؛كما الحاجة (مثل graphic) إلى إخراج كميات، إلى إخراج الكاميرا، إلخ.</td>
<td>استخدم على (مثل graphic) عدد مرات (مثل graphic) كما العدده ؛كما الحاجة (مثل graphic) إلى إخراج كميات، إلى إخراج الكاميرا، إلخ.</td>
<td>53.02</td>
</tr>
<tr>
<td>لا تدخل رقم الطوارئ لمنطقك الحالية (مثل: 011 أو 12 location)</td>
<td>لا تدخل رقم الطوارئ لمنطقك الحالية (مثل: 011 أو 12 location)</td>
<td>43.68</td>
</tr>
<tr>
<td>إخراج رقم الطوارئ من مكان إلى آخر</td>
<td>location</td>
<td>25.64</td>
</tr>
<tr>
<td>عمل (ثلgraphic) في وضع (مثل graphic) في وضع</td>
<td>عمل (ثلgraphic) في وضع (مثل graphic) في وضع</td>
<td>50.00</td>
</tr>
<tr>
<td>إذا كانت بعض الخصائص في الاستخدام، فقد نحتاج إلى حلها قبل أن تكن من إجراء إعداد طارئ</td>
<td>إذا كانت بعض الخصائص في الاستخدام، فقد نحتاج إلى حلها قبل أن تكن من إجراء إعداد طارئ</td>
<td>38.14</td>
</tr>
<tr>
<td>راجع هذا النيل مع وكيل الخدمة المحلية لخدمات</td>
<td>راجع هذا النيل مع وكيل الخدمة المحلية لخدمات</td>
<td>55.36</td>
</tr>
<tr>
<td>عند إجراء إعداد طارئ</td>
<td>عند إجراء إعداد طارئ</td>
<td>38.37</td>
</tr>
<tr>
<td>إذا بالمعلومات كافية ودقيقة.</td>
<td>إذا بالمعلومات كافية ودقيقة.</td>
<td></td>
</tr>
<tr>
<td>إذا أخذت حقن الهواتف هو رمزي الوصلة الواحدة لتصال</td>
<td>إذا أخذت حقن الهواتف هو رمزي الوصلة الواحدة لتصال</td>
<td>37.40</td>
</tr>
<tr>
<td>إذا أخذت حقن الهواتف هو رمزي الوصلة الواحدة لتصال</td>
<td>إذا أخذت حقن الهواتف هو رمزي الوصلة الواحدة لتصال</td>
<td>48.72</td>
</tr>
<tr>
<td>تأكد من إجراء إعداد طارئ، أعمله على الموديل النا</td>
<td>تأكد من إجراء إعداد طارئ، أعمله على الموديل النا</td>
<td>71.17</td>
</tr>
<tr>
<td>nokia pc sync</td>
<td>nokia pc sync</td>
<td></td>
</tr>
<tr>
<td>Model Type</td>
<td>Central Processing Unit</td>
<td>GPU</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>General-Purpose (4-core)</td>
<td>2.0 GHz</td>
<td>R5</td>
</tr>
<tr>
<td>General-Purpose (8-core)</td>
<td>3.2 GHz</td>
<td>R7</td>
</tr>
<tr>
<td>High-Performance (12-core)</td>
<td>4.0 GHz</td>
<td>R9</td>
</tr>
<tr>
<td>Extreme Performance (16-core)</td>
<td>5.0 GHz</td>
<td>R15</td>
</tr>
</tbody>
</table>

*Prices are subject to change and may vary by region.*
<table>
<thead>
<tr>
<th>Human Translator</th>
<th>Our Translation</th>
<th>Edit Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ أثناء استخدام حاسة اليومية الأصلية ، حاول الركوب كأتي خلف الكتف .</td>
<td>عند ساعة الأدنى ، في الوضع ، حاول الركوب كأتي خلف الهوائي الأعلى .</td>
<td>55.88</td>
</tr>
<tr>
<td>$ إذا كان هذا الركوب ، كأتي خلف يابسة الخاصة ، يعمال</td>
<td>هذا الركوب ، كأتي خلف يابسة الخاصة ، يعمال باستخدام الإشارات الإسارية .</td>
<td>97.12</td>
</tr>
<tr>
<td>$ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية</td>
<td>إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .</td>
<td>94.62</td>
</tr>
<tr>
<td>$ طباخة يالقية وطباخة يابسة في كل الأحوال .</td>
<td>طباخة يالقية وطباخة يابسة في كل الأحوال .</td>
<td>90.74</td>
</tr>
<tr>
<td>$ الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>90.74</td>
</tr>
<tr>
<td>$ الطباخة يالقية وطباخة يابسة في كل الأحوال .</td>
<td>الطباخة يالقية وطباخة يابسة في كل الأحوال .</td>
<td>90.74</td>
</tr>
<tr>
<td>$ طباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>طباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>90.74</td>
</tr>
<tr>
<td>$ الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>90.74</td>
</tr>
<tr>
<td>$ الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>90.74</td>
</tr>
<tr>
<td>$ الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>90.74</td>
</tr>
<tr>
<td>$ الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>الطباخة يابسة وطباخة يالقية في كل الأحوال .</td>
<td>90.74</td>
</tr>
</tbody>
</table>

الترجمة الأمثلة:

1. $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأت جانب حاوتة خلف الهوائي الأعلى .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإسارية .
   $ إذا كنت تكون كأتي خلف كاتب البسمة ، والعديد من الإشارات الإساري
وظائف ملتقط الهاتف بالاختلاف

$ wap$

$ مشاريع مراقبة الهاتف بالاختلاف$

$ مشاريع دفع تكاليف مشتريات بواسطة$ 65.22

$ قد يسمح المحمي لعدة 10 دقائق تقريباً أو$ 67.33

$ حتى يتم توصيل سماعة الرأس جاف فوق متوافق ،$ 68.00

$ أيهما أقرب .$ 100.00

$ الهاتف :$ 52.58

$ يجب عليك توفير تفاصيل الحساب الصحيح .$

$ الهاتف المحمول الخاص بالعمل$ 100.00

$ إزالة المسار ( المسارات ) الحدود / أو قائمة$ 100.00

$ أو قائمة ( قوانين ) التشغيل من الهاتف .$

$ generaltagpurpose fourtagwire ( interface = g 4 wif )$

$ كواجهة رباعية الأسلاك ذات عرض عام ()$ 62.77

$ central processing unit ( المركزية$ 96.08

$ راهنة المعالجة المركزية$ 100.00

$ اضغط على زر التحرير عند أسفل الراديو لتخليص إلقاء$ 85.29

$ اختر لفتة هاتف البطارية$ 85.29

$ إذا توفر لدينا أكثر من رقم محفوظ تحت الإسم ،$ 77.17

$ فاتصل إلى الرقم المطلوب واضغط على " موافق " .$ 77.17