Chapter 10: Automatic Rule Induction in Arabic to English Machine Translation Framework

Khaled Shaalan[ab], Ahmad Hany Hossny[c]

[a] *Fellow, School of Informatics, University of Edinburgh, UK*
[b] *Faculty of Informatics, The British University in Dubai, P O Box 345015, Dubai, UAE*
[c] *Centre for Intelligent Systems Research, Deakin University*
Pigdons Rd, Waurn Ponds, Vic, 3217, Australia
Khaled.shaalan@buid.ac.ae, Ahossny@deakin.edu.au

Abstract

This paper addresses exploiting a supervised machine learning technique to automatically induce Arabic-to-English transfer rules from chunks of parallel aligned linguistic resources. The induced structural transfer rules encode the linguistic translation knowledge for converting an Arabic syntactic structure into a target English syntactic structure. These rules are going to be an integral part of an Arabic-English transfer-based machine translation. Nevertheless, a novel morphological rule induction method is employed for learning Arabic morphological rules that are applied in our Arabic morphological analyzer. To demonstrate the capability of the automated rule induction technique we conducted rule-based translation experiments that use induced rules from a relatively small data set. The translation quality of the hybrid translation experiments achieved good results in terms of WER.

Keywords: transfer rule induction, transfer-based machine translation, morphological rule induction, inductive logic programming, Arabic-to-English machine translation.

1. Introduction

Machine translation systems incorporate two main components: the translation model and the translation engine. The translation model follows empirical (statistical or machine learning) approaches, linguistic (rule-based) approaches, or a hybrid of both. The translation engine utilizes the translation model to transform a source sentence into a target sentence. In empirical approaches, this is based on finding the most probable translation of a sentence using data gathered from an aligned bilingual corpus. Hence, some researches refer to these approaches as corpus-based. In linguistic approaches, linguistic knowledge is used in the representation of translation units. These approaches are based on linguistic analysis of the source sentence and generation of the target sentence. If the level of the analysis is not deep enough to produce a format suitable for the generation of the target language, a collection of transformations is applied to the syntactic analysis of the source language in order to construct a target-language syntactic representation. Historically, many Arabic-English machine translation systems, including commercial ones, are transfer-based. The advantages of this approach are that it can attain high performance but at the expense of the large efforts needed to build the necessary linguistic resources (Abdel Monem et al., 2008). On the other hand, corpus-based approaches require a very large parallel corpus that is neither easily available nor affordable. Nowadays, there is a growing interest in hybrid approaches. For example, a proposed approach to statistical machine translation (SMT) that combines ideas from phrase-based SMT (Koehn et al., 2003) and traditional ruled-based grammar generation (Riezler and Maxwell, 2006) provides significant improvements in the grammaticality of translations over state-of-the-art phrase-based SMT on in-coverage examples, suggesting a possible hybrid framework.

In this research, we address exploiting a supervised machine learning technique to develop an example-based transfer tool that automatically induces Arabic-to-English syntactic transfer rules from chunks of parallel linguistic resources. The Arabic-English language pair is very different in morphology and syntax. The richness of Arabic morphology has led us to develop a novel morphological rule induction method for learning Arabic morphological rules that are applied in our Arabic morphological analyzer. To demonstrate the capability of these automated techniques we conducted a rule-based translation experiments that use induced rules from a relatively small data set. The induced syntactic transfer rules are learned from a set of Arabic-English example pairs, each of which includes the feature structure representing the linguistic knowledge of each word. The obtained results from the evaluated hybrid system, i.e. rule-based translation system whose transfer rules has been generated using inductive techniques, were promising and assure that using the proposed machine learning technique would improve the performance of the Arabic-to-English rule-based sentence translation.

The rest of the paper is organized as follows. Section 2 gives a brief background on inductive logic programming with regard to natural language processing tasks. Section 3 discusses some hybrid machine translation systems. Section 4 describes the related research on automatic rule induction. Section 5 introduces the Arabic-English transfer rules induction technique. Section 6 reports results from rule-based translation experiments that use induced rules from a relatively small data set. Section 7 gives some concluding remarks with direction for future work.

2. Inductive Logic Programming

Inductive Logic Programming (ILP) (Muggleton, 1999) is a machine learning approach that uses logic programming as a representation for examples, background knowledge, and hypotheses. Given an encoding of the known background knowledge and a set of examples represented as a logical database of facts, an ILP system derives a hypothesized logic program which entails all the positive and none of the negative examples.

ILP works on three steps: 1) analyze input and output data according to previous experience of the subject to get some kind of abstraction on both source and target in order to be mapped later, 2) train the system positively to identify which input should be mapped to which output as logical production rules, and 3) train the system negatively to identify which inputs should never map to which output.

In order to be able to use ILP in natural language processing (NLP), in particular machine translation, we have to treat the sentence as a structure or pattern, and identify the linguistic features (i.e., feature structure representing the linguistic knowledge of each word) to be evaluated and unified between the left-hand side and right-hand side of the induced rule. The system should be trained positively on a valid set of input-output example pairs. The system also should be trained negatively on a set of non-valid example pairs in order to induce the negative rules that will act as excluding rules. Negative training for NLP tasks is hard as there usually are an infinite number of language combinations that does not match on the lexical, morphological, syntactic and semantic level.

The main advantage of ILP, in general, is that it provides logarithmic growth for first order learning (Koriche, 2005). This leads to system consistency after a limited set of trainings, i.e. degradation of newly generated rules. ILP has been successfully implemented in NLP tasks like morphological analysis (Hossny et al., 2008), Part of Speech (POS) tagging (Eineborg et al., 1999), POS disambiguation (Lindberg et al., 1999), and machine translation (Hossny et al., 2009). In this paper, we address exploiting ILP to automatically induce Arabic-to-English rules from chunks of parallel linguistic resources.

3. Hybrid machine translation systems

Our focus is on the development of a hybrid machine translation system. The system is based on an Arabic-to-English rule-based machine translation approach that uses induced rules in its transfer and analysis steps. An example-based machine learning approach, which is the main contribution of the work presented here, is used to induce the Arabic-to-English transfer rules from a set of example pairs.

In the rest of this section, we briefly describe the notable hybrid machine translation systems for various languages that were developed from rule-based machine translation (RBMT), example-based machine translation (EBMT), and statistical-based machine translation (SMT) approaches.

3.1. RBMT/EBMT Hybrids

Shirai et al. (1997) have proposed a method that gathers the strength points in both of RBMT and EBMT for English-Japanese Translation. The algorithm was briefly described in three steps 1) Select a set of candidate sentences which are similar to the input sentence, 2) Select the most typical translation out of those corresponding to the candidates, and 3) Use this translation and its source as templates to translate the input sentence. By discarding candidates with a typical translation, the algorithm filters out free, incorrect or context dependent translations.

Carl et al., (1998) described an NLP example-based translation application called Case-Based Analysis and Generation Module (CBAG), which is applied within a conventional RBMT system to drastically improve its performance. The main idea behind the CBAG module is to introduce a significant share of human translation experience accumulated in Translation Memories, which are, after all, relatively simple but very large and accurate collections of bilingual texts. They applied this application on English-French and English-German translation, and they could determine the types of word combinations, and chunks, introduced into a case base, that have a positive and sizeable effect on the translation quality and performance, and they could state the induction mechanisms to be used to extend the case base without creating additional noise.

3.2. RBMT/SMT Hybrids

Ambati et al., (2007) have presented a hybrid EBMT/SMT approach to perform translation from English to Hindi. They performed matching by considering the longest match of the input sentence available in the example database, and performed the alignment using a manual and a statistical dictionary build from GIZA++ and the best Viterbi alignment given by GIZA++ for each sentence pair in the example database. Finally, the combination is done simply by merging different translated fragments to obtain the complete translated sentence.

Chen et al. (2007) have proposed an architecture that allows combining SMT with RBMT in a multi-engine setup. It uses a variant of standard SMT technology to align translations from one or more RBMT systems with the source text. They incorporated phrases extracted from these alignments into the phrase table of the SMT system and used the open-source decoder MOSES to find good combinations of phrases from SMT training data with the phrases derived from RBMT.

3.3. EBMT/SMT Hybrids

Imamura et al. (2004) proposed an EBMT method based on syntactic transfer, which selects the best translation by using models of SMT. This method is roughly structured using two modules. The first is an example-based syntactic transfer module which constructs tree structures of the target language by parsing and mapping the input sentence while referring to transfer rules. The other module is a statistical generator, which selects the best word sequence of the target language in the same manner as SMT. Therefore, this method sequentially combines EBMT and SMT. The proposed method has the advantages of improving the quality of machine translation by selecting the best translation from the similarity judgment between the input sentence and the source part of the examples. The other advantage is making the search space smaller as the example-based transfer generates syntactically correct candidates for the most appropriate translation.

Sumita et al. (2004) conducted a project called Corpus-Centered Computation (C3). C3 places corpora at the center of its technology. Translation knowledge is extracted from corpora. Translation quality is gauged by referring to corpora; the best translation among multiple-engine outputs is selected based on corpora. The corpora themselves are paraphrased or filtered by automated processes to improve the data quality on which translation engines are based. This proposes two endeavors that are independent: 1) a hybridization of EBMT and statistical models, and 2) a new approach for SMT, phrase-based Hidden Markov Model (HMM). The hybridization was used in the “unrestricted” Japanese-to-English track while the phrase-based HMM was used in the “supplied” Japanese-to-English and Chinese-to-English tracks.

Aramaki et al., (2005) proposed a probabilistic language model, which deals both the example size and the context similarity. The conducted experiments show tha, the proposed model has achieved a slightly better translation quality than the state-of-the-art EBMT systems. The proposed algorithm consists of two modules: alignment and translation modules. The alignment module builds translation examples from a corpus in three steps: 1) conversion into phrasal dependency structures, 2) alignment of phrases using a translation dictionary, and 3) building translation examples database. The translation module generates a translation through three steps: 1) input sentence analysis 2) select the closed translation examples, and 3) target sentence generation.

4. Related work

In this section we address the related research on automatic rule induction from a data set of example pairs using different techniques for various languages.

A Bootstrapping, Template- Driven Approach to Example-Based MT

Veale et al. (1997) built a system, so-called Gaijin, which implements a template-driven approach to example-based English-German machine translation. Gaijin is a system that employs statistical methods, string-matching, case-based reasoning and template-matching to provide a linguistics-lite EBMT solution. The only lingual input needed by this system is psycholinguistic constraint - the marker hypothesis - that is minimal in size and simple to apply. The system consists of six sequential steps: 1) bilingual corpora alignment, 2) automatic lexica construction, 3) transfer-template generation, 4) example retrieval, 5) example adaptation, and 6) new example acquisition. The system generates template matching rule mapping from source to target like this:

```
Template (24, English, German,
[s(A, prep , a24), s(B, det , b24), s(C, prep , c24), s(D, pro , d24),s(E, prep , e24) ],
[t(A, _, a24), t(B, det , b24), t(C, prep , c24), t([D|E], prep , [d24,e24])] ).
```

Example-Based Machine Translation of the Basque Language

Stroppa et al. (2006) presented a Data-Driven machine translation system which exploits both EBMT and SMT techniques to extract a data set of aligned (Basque-English) chunks. For the extraction of the EBMT data resources, they made use of two different chunking methods. In the case of English, they employed a marker-based chunker that depends on the marker hypothesis (Green, 1979). For Basque, they used the dedicated tools developed at the University of the Basque Country while investigating the application of the marker-based chunker to Basque. The chunks are then aligned using a dynamic programming algorithm which is similar to an edit-distance algorithm while allowing for block movements (Leusch et al., 2006). This aligner also relies on relationships between chunks, which they compute in several ways.

Learning Transfer Rules for Machine Translation from limited data

Lavie et al. (2004) proposed a machine translation approach that is specifically designed to enable rapid development of machine translation for languages with limited amounts of online resources. Their approach assumes the availability of a small number of bi-lingual speakers of the two languages, but these need not be linguistic experts. The bi-lingual speakers create a comparatively small corpus of word aligned phrases and sentences (on the order of magnitude of a few thousand sentence pairs) using a specially designed elicitation tool. From this data, the learning module of the system automatically infers hierarchical syntactic transfer rules, which encode how syntactic constituent structures in the source language transfer to the target language (Probst et al., 2002). The collection of transfer rules is then used in the run-time system to translate previously unseen source language text into the target language. They reported results from experiments, which under the very limited data training scenario they constructed their

XFER system with all its variants has significantly outperformed a SMT system on Hindi-to-English machine translation.

5. An Arabic-English transfer rules induction technique

In this section, we describe our experience on how we successfully constructed a rule-based translation model using inductive logic programming (ILP) (Muggleton, 1999) to learn and induce Arabic-to-English transfer rules from chunks of a Linguistic Data Consortium's (LDC) parallel corpus (Arabic Treebank with English translation-LDC2005E46). The rule induction process consists of four main steps: a) Word-to-Word alignment of sentence pairs extracted from the parallel corpus, b) Partition each sentence into chunks, c) Determine the patterns and feature structure representing the linguistic knowledge of each word using our morphological analysis tool, and d) Induce translation rules by identifying, for each rule, the left-hand side (LHS) and the right-hand side (RHS) patterns and construct the link between them.

5.1. Word-to-Word alignment

In order to exploit a parallel text, some kind of text alignment, which identifies equivalent text segments of source and target translations, is a prerequisite for rule induction. This step is similar to the alignment produced by Giza++ (Och et al., 2000). Figure 1 shows the word-to-word alignment of Example 1. It shows one-to-one and one-to-many word alignment.

Example 1: *Arabic-English word alignment of a parallel sentence*

Source Index	S1	S2	S3	S4	S5	S6	S7		
Source (S)	هزم	القطن	بطل	الكاميرون	من	الأهلي	المصري		
Alignment	5, 6	4	3	1, 2	7	9	8		
Target (T)	The	Cameron's	champion	"Cotton"	was	defeated	by	Egyptian	Ahly
Target Index	T1	T2	T3	T4	T5	T6	T7	T8	T9
Transliteration	huzima	Al.quT.nu	baTal	AlkaAmiyaruwn	min.	Al.Âah.liy~	Al.miS.riy		

We randomly selected 300 sentences from the LDC parallel corpus (LDC2005E46) in order to prove the capability of our machine learning approach in automating the induction of transfer rules from relatively small training data set of Arabic-English example pairs.

<figure 1 goes about here>

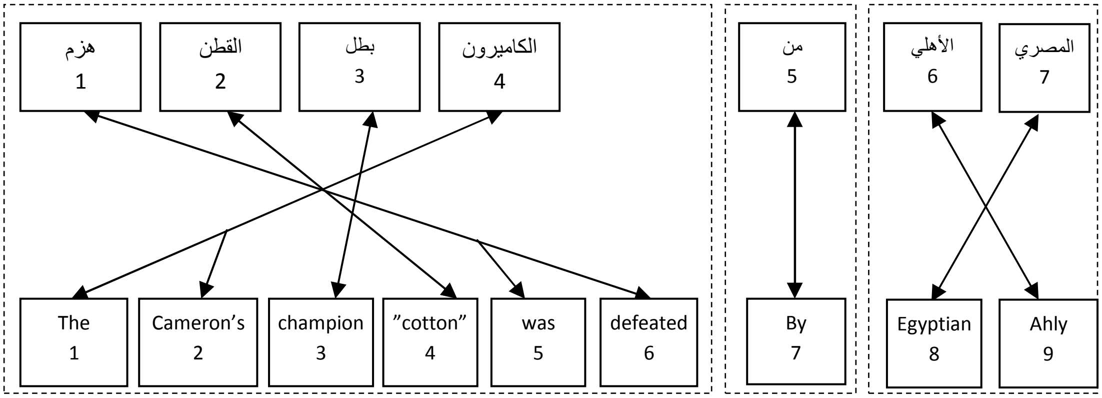

Fig. 1: Arabic-English word alignment for a complex sentence leads to partitioning it to multiple chunks.

5.2. Sentence partitioning

Sentence partitioning (chunking) is an important step in our induction process for two reasons: 1) It simplifies units used in induction by dealing with a phrase or chunk rather than an entire sentence which is sometimes has a complex structure (Roh et al. 2001), and 2) induces a set of transfer rules per sentence instead of one which would result in a rich set of induced transfer rules with better coverage. These reasons have the impact of giving the transfer module of the intended machine translation system the opportunity to operate on rules at different granularity.

We perform partitioning of both the source and target sentence into a sequence of chunks based on word alignment dependency such that there is no overlap of word ordering between chunks. This is depicted in Figure 1. We used *Find Chunk Boundary* algorithm, see Figure 2, to detect the chunk boundaries, i.e. identify where we split the sentence into chunks, which is similar to the phrase extraction algorithm given by Koehn et al. (2003).

By applying the *Find Chunk Boundary* algorithm to Example 1, we get three chunks as shown in Figure 3. The identification of the first chunk is as follows. The algorithm starts by the source node S1, finds its target node T6, then traces backward the target sentence till its first node T1, and then finds its associated source node S4. It again traces backward the source sentence till the first node S1 of this chunk, i.e. the starting source node. This two-way process is repeated which results in two more chunks.

5.3. Arabic morphological analysis

Determining the morphological analysis by recognizing the feature structure representing the linguistic knowledge of each source word form is an important step before the actual induction of transfer rules takes place. The feature structure consists of a *feature:value* pair (e.g. gender:feminine, tense:perfect, and number:dual). During the course of transferring a source Arabic word into a target English word, these features might be carried over or modified, which affects the generation of the inflected forms of the target language.

We developed an Arabic morphological analyzer that applies morphologically rules induced from monolingual parallel annotated example pairs. The morphologically induced rules are generated by our Automatic Morphological Rule Induction Tool (AMRIT). As shown in Figure 4, AMRIT induces morphological rules from monolingual example pairs of inflected forms and their stems.

The AMRIT module compares each example pair for their both vocal patterns and feature structures representations in order to induce the rule that causes the morphological changes. The vocal patterns are representations of the Arabic word consisting of a sequence of vowels and consonants (cf. Beesley, 1996). An important feature of AMRIT is that it is able to automatically acquire both regular and irregular Arabic morphological analysis rules. Irregular forms such as weak verbs and broken plural are very hard to analyze as well as marked correctly when generating annotated tagged Arabic corpora. This is better explained by the following example.

<Figure 2 goes about here> <Figure 3 goes about here>

```
Algorithm: Find Chunk Boundary
Input:     Ss: Source Sentence, Ts: Target Sentence, source_end: Source Chunk End Index,
           target_end: Target Chunk End Index
Parameters initial values:
           source_end = source chunk start position,
           target_end = target chunk start position,
Output: Index of the end of the chunk

  src_max_idx ← max(index of source_end)
  FOR each word target_word in target sentence Ts let target_idx be the index of target_word (where the target_idx is less than
  target_end)
           src_idx ← index (word associated with target_word in source sentence)
           IF  Src_max_idx < src_idx THEN let Src_max_idx ← src_idx END IF
  END FOR
  target_end_word  ← word at the target_end position in the target sentence Ts
  IF (src_max_idx > source_end) THEN
           let new_target_end ← FindChunkBoundary (Ts, Ss, target_end, src_max_id )
           IF new_target_end == target_end THEN
                     Return src_max_idx  as the source chunk boundary.
           ELSE
                     Return new_src_idx as the index of the word associated with new_target_word in source sentence.
           END IF
 ELSE
           Return source_end
 END IF
```

Fig. 2: An Algorithm for finding the chunk boundary from an aligned sentence pair

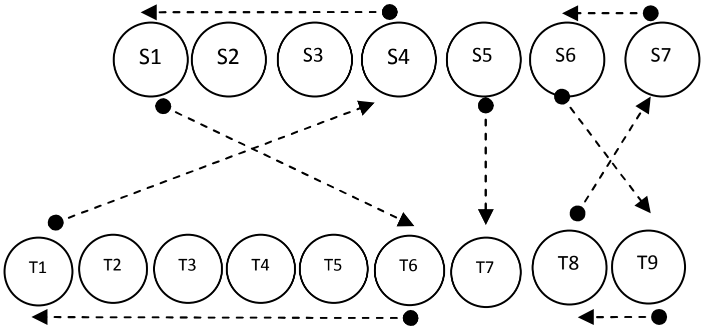

Fig. 3: Two-way trace for detecting chunk boundaries for the three chunks

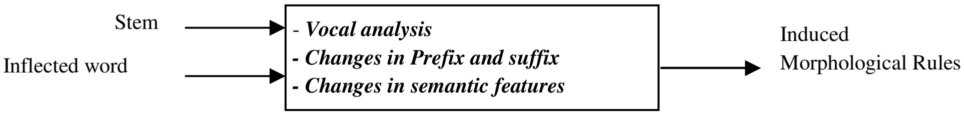
Stem
Inflected word
- Vocal analysis
- Changes in Prefix and suffix
- Changes in semantic features
Induced
Morphological Rules

Fig. 4: Morphological Rule induction process using AMRIT

Example 2*: A morphological analysis rule for analyzing the imperfect assimilated (weak initial radical) verb into its stem.*

Consider, the stem “وقى” /waqaýa/ (to-protect) that has the vocal representation “VCV”. In imperfect form, it becomes “يقى” /yaqiy./ that is produced by removing the first weak radical letter “و” (Waw) and adding the imperfect tense prefix letter “ي” (Yeh). The induced rule that is generated by AMRIT from these two words is as follows:

يقى → *وقي*
V2C1 ي (tense:imperfect) → *V2C1 و (tense:perfect)*
where V stands for vowels and C stands for consonants

Applying this rule by our Arabic morphological analyzer to analyze the imperfect form "يعى" /yaçiy./ (is-conscious), which has the vocal representation (VC ي), yields the stem "وعى" /waçaýa/ (to-be conscious), which has the vocal representation (VCو). Moreover, AMRIT is able to induce complex rules, e.g. the following rule analyzes the word 'يهتدون' /yah.taduw.n/ (discover-/guide-they [pl,imperfect,masculine]) into the stem 'إهتدى' /Ăih.tadaý/ (discovered-/guided-he [sg,perfect,masculine]).

ونC3C2C1 ي(tense:imperfect,gender:Masculine,number:plural,person:Third)
→ ى+ون C3 C2 C1 إ
(tense:perfect,gender:Masculine,number:singular,person:Third)+ي

To demonstrate the capability of our Arabic morphological analyzer in analyzing a sentence using induced rules by AMRIT, consider the verbal sentence "تقتدي الفرق بالأبطال" /taq.tadiy Al.far.qu biAlÂab.TaAl/ (the teams take the champions as role model). The verb "تقتدي" /taq.tadiy/ (take-as-role-module –an imperfect form of the stem إقتدى /Ăiq.tadaý/) will be analyzed exactly in a way similar to the verb "يهتدي" /yah.tadiy/. The noun "الفرق" /Al.far.q/ (teams—a broken plural of the noun فرقة /fir.qaħ/) will be analyzed exactly in a way similar to the noun "البرك" /Al.bir~ak/ (bonds— a broken plural of the noun "بركة" /barakaħ/). The noun "الأبطال" /Al.Âab.TaAl/ (champions—a broken plural of the noun "بطل" /baTal/) will be analyzed exactly in a way similar to the noun "الأعطال"/Al.Âaç.TaAl/ (malfunctions— a broken plural of the noun "عطل" /çuTil/).

To sum up, our Arabic morphological analyzer uses the induced rules to analyze each word in the input Arabic sentence to generate its stem and feature structure representing the linguistic knowledge of this word. This results in a representation suitable for rule induction which consists of a sequence of chunks, each of which consists of a sequence of morphologically analyzed words with their linguistic features.

5.4. Transfer Rule Induction

5.4.1. Rule construction

The transfer rule construction process involves establishing a mapping from a sequence of source chunks (LHS) to a sequence of target chunks (RHS). We recall that a chunk itself is a sequence of words. Each word is represented by a feature structure that encodes its linguistic knowledge. So, a transfer rule maps the source linguistic knowledge into a target linguistic knowledge.

The rule construction process considers the induction of rules at varying grain size in order to allow for selecting the applicable rule that matches the input pattern. This gives the induced transfer rules better coverage. In the following we, show how transfer rules are induced from chunks shown in Figure 1.

So far, we have three source chunks [S1 + S2 + S3 + S4], [S5] and [S6 + S7] linked with three target chunks [T1 + T2 + T3 + T4 + T5 + T6], [T7] and [T8 + T9], respectively. As shown in Figure 5, these chunks can be used to generate six induced rules: 1) one rule composed of a sequence of three chunks, 2) two rules composed of a sequence of two chunks, and 3) three rules composed of a sequence of one chunk. In general, the number of rules generated per source sentence is the sum of a number series 1 to *n*, where *n* is the number of chunks generated from this sentence:

$$\sum_{i=0}^{n} i = 1+2+\ldots+n = (n*(n+1))/2$$

Each term *i* in this series forms a group of rules consisting of i rules, each of which links a sequence of source *i* chunks with a sequence of target *i* chunks.

<Figure 5 goes about here>

1 rule with 3 chunks:
Rule1: [S1 + S2 +S3 + S4] + [S5] + [S6 +S7] → [T1 + T2 + T3 + T4 + T5 + T6] + [T7] +[T8 +T9]

2 rules with 2 chunks each:
Rule2: [S1 + S2 +S3 + S4] + [S5] → [T1 + T2 + T3 + T4 + T5 + T6] + [T7]
Rule3: [S5] +[S6 +S7] → [T7] +[T8 +T9]

3 rules with 1 chunk each:
Rule4: [S1 + S2 +S3 + S4] → [T1 + T2 + T3 + T4 + T5 + T6]
Rule5: [S5] → [T7]
Rule6: [S6 +S7] → [T8 +T9]

Fig. 5: Three chunks can generate up to six induced rules.

5.4.2. Feature Unification

After the rule structure is determined, the *feature:value* pairs of each word in both sides (Arabic–English) of the induced rule is unified using Unification Based Grammar formalism. This unification process will result in: setting specific feature's value constraint (a constant represented by symbols that has initial lower case letter), determining a variable to be unified during the machine translation process (represented by an identifier with an initial upper case letter) in order to apply carried over source linguistic constraint, and generating anonymous variable (underscore symbol to suppress irrelevant value). For example, consider the following transfer rule.

{number:S1,gender:masc ,cat:verb, tense:P1} → *{number:S1, gender:_ , cat:verb, tense:P1}*

This rule says that in order to transfer a source Arabic verb into a target English verb, the gender of the source verb should be set to masculine, and both source and target verbs should be mapped to the same number and tense. Notice that the gender of the English verb is neither constrained to a specific nor carried over. So, it is set to anonymous value.

5.5. An example of a transfer rule induction

In this section, we give an example of a transfer rule induction. The steps specified above are applied on an aligned Arabic-English chunk pair to induce the rule shown in Figure 6. Then, we demonstrate how the induced rule can be used by the transfer module of a machine translation system to transfer a similar source Arabic input into a target English output.

Chunk 1 = هزم الأهلي المصري
Chunk 2 = The Egyptian al-Ahly defeated

Arabic-English word alignment:
هزم + الأهلي + المصري → The Egyptian+ al-Ahly + defeated
Wa[1] + Wa[2] + Wa[3] → The We[3] + We[2] + We[1]

Arabic morphological analysis:

```
هزم                 (Wa[1])  {number:singular, gender:masc,cat:verb,tense:perfect}
Defeated   (We [1])  {number:singular,gender:masc ,cat:verb,tense:perfect}

الأهلي                (Wa[2])  {number:singular,gender:_,sub_cat:propernoun}
al-Ahly    (We[2])  {number:singular,gender:_,sub_cat:propernoun}

المصري     (Wa[3])  {number:singular,gender:masc,sub_cat:adj,
                              definite_article:yes}
The                          {definite_article:yes}
 Egyptian  (We[3])  {number:singular,gender:masc,sub_cat:adj}

Constructing an inductive rule:
Wa[1]      {number:singular,gender:masc,cat:verb,tense:perfect}              +
Wa[2]      {number:singular,gender:_,sub_cat:propernoun}                     +
Wa[3]      {number:singular,gender:masc , sub_cat:adjective, definite_article:yes} →
The        {definite_article:yes}                                            +
We[3]      {number:singular,gender:masc , sub_cat:adj}                       +
We[2]      {number:singular,gender:_,sub_cat:propernoun}                     +
We[1]      {number:singular,gender:masc,cat:verb,tense:perfect}

Unifying LHS & RHS sides of the inductive rule
Wa[1]      {number:S1,gender:masc,cat:verb,tense:P1}                         +
Wa[2]      {number:S2,gender:_ ,sub_cat:propernoun}                          +
Wa[3]      {number:C3,gender:S3,sub_cat:T3,definite_article:yes} →
The        {definite_article:yes}                                            +
We[3]      {number:C3,gender:_,sub_cat:T3}                                   +
We[2]      {number:S2,gender:_,sub_cat:propernoun}                           +
We[1]      {number:S1, gender:_,cat:verb,tense:P1}
```

Fig. 6: An example of constructing Arabic-English transfer rule from an aligned chunk
<Figure 6 goes about here>

Figure 7 shows the results of applying the transfer rule induced in Figure 6 to the first aligned chunk of the source Arabic sentence “حققت فينوس البطلة رقم قياسي جديد” /Haq~aqt fiy.nuws Al.baTalah̄ raq.m qiyaAsiy jadiy.d/ in order to transfer it to a format suitable for generating the target English sentence.

```
Arabic Sentence  = حققت  فينوس البطلة رقم قياسي جديد
English Sentence  = [The champion Venus achieved new record]

Sentence partitioning:
Chunk1 = حققت  فينوس البطلة        → The champion Venus achieved
Chunk2 = رقم قياسي جديد            → new record

Arabic-English word alignment:
حققت + فينوس + البطلة → The champion + Venus + achieved
Wa[1]  +  Wa[2] +  Wa[3]  →  The We[3]    + We[2]  +  We[1]

Arabic Morphological analysis:
حققت       (Wa[1])    {number:singular, gender:fem, cat:verb, tense:perfect}
فينوس      (Wa[2])    {number:singular, gender:fem , sub_cat:propernoun}
البطلة      (Wa[3])    {number:singular, gender:fem , sub_cat:adj, definite_article:yes}

Matching with the induced rule:
Wa[1]      {number:S1,gender:masc, cat:verb, tense:P1}             +
Wa[2]      {number:S2,gender:_ , sub_cat:propernoun}               +
Wa[3]      {number:C3,gender:S3, sub_cat:T3, definite_article:yes}
→
The        {definite_article:yes}                                  +
We[3]      {number:C3,gender: _,sub_cat:T3}                        +
We[2]      {number:S2,gender:_ ,sub_cat:propernoun}                +
We[1]      {number:S1,gender:_ ,cat:verb,tense:P1}

Transfer output:
The                             {definite_article:yes}
We[3] = champion      {number:singular, gender:_, sub_cat:adj}
We[2] = Venus          {number:singular, gender:_ , sub_cat:propernoun}
```

We[1] = Achieved {number:singular, gender:_ , cat:verb,tense:perfect}

Fig. 7: An example of applying Arabic-English induced transfer rule

<Figure 7 goes about here>

6. Experiments

To evaluate our inductive machine leaning technique, we conducted two related experiments using considerably small dataset. The objective of the first experiment is to measure the effect of the number of training examples on Arabic-English transfer rule induction. The objective of the second evaluation experiment is to assess the translation performance by comparing the automatically translated sentences with a gold standard reference test data. Fig 8 shows an illustration of the scheme used to conduct these two experiments.

<Figure 8 goes about here>

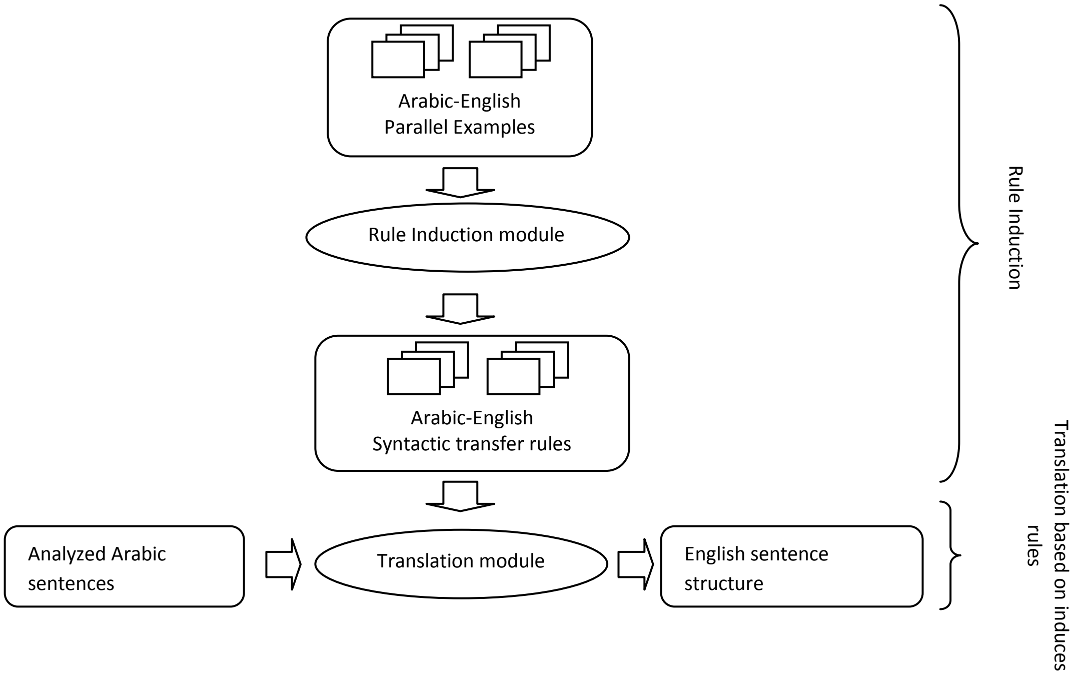

Fig. 8: Experiments to evaluate two modules: rule induction and machine translation module built on top of it.

6.1. Rule induction experiment

A set of 300 parallel sentences was used as a test set. The average sentence length is 10 words. This set was randomly chosen from the Arabic Treebank with English translation (LDC2005E46). We applied our sentence partitioning (chunking) algorithm on the 300 parallel aligned examples, which produced 2087 chunks with the frequency distribution of 1 to 10 chunk size shown in Table 1.

Table 1. Frequency Distribution of chunks

Chunk size	1	2	3	4	5	6-7	8-10	Total
Frequency	1498	304	151	81	32	15	6	2087

<Table 1 goes about here>

The rule induction of these chunks has produced 1115 unique Arabic-to-English transfer rules. Figure 9 shows that the growth rate of the number of induced rules proportional to the number of chunks takes a logarithmic shape. This observation indicates that our induction technique is capable to induce from a relatively small dataset the most frequently used Arabic-to-English transfer rules.

<Figure 9 goes about here>

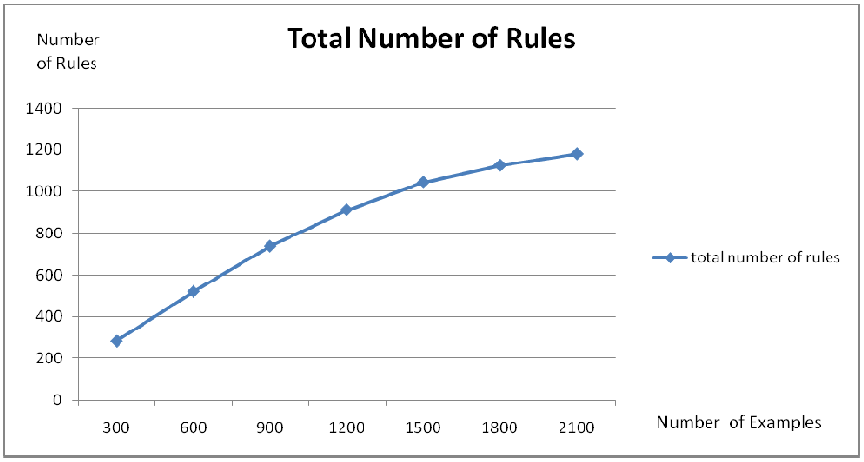

Fig. 9: Growth rate of the total number of induced rules

From the analysis of the results of this experiment we observe the following about the similarity of the induced rules that affects the growth rate of the newly induced rules; the higher similarity the lower newly induced rules and the vice versa. In our experiment, we found that there are two factors affecting the similarity of newly induce rules: Part of speech (POS) and chunk size. As the number of POS's increases, the probability of rules similarity decreases. Moreover, increasing the number of words in a chunk leads to a lower probability of similarity.

6.2. Translation Performance experiment

A second experiment that uses another 180 dataset from LDC2005E46 is conducted. The objective is to test the performance of a hybrid rule-based Arabic-English machine translation system that is built on top of the induced syntactic transfer rules produced by the previous experiment. We employed Word Error Rate (WER) metric which has been used in evaluating machine translation when the word sequence output can have a different length from the reference word sequence. The results depend on the minimum chunk size (i.e., rule size) and the sentence size (number of rules used in the translation process). Table 2 shows the effect of chunk size on both of coverage and error rate. From this table we observe that the minimum chunk size is in a reverse proportional to both the coverage and error ratio.

Table 2. The effect of chunk size on coverage and error rate

Minimum chunk size	Coverage	Error rate
1	100%	88%
2	96%	68%
3	80%	39%
4	54%	24%
5	23%	11%

<Table 2 goes about here>

7. Conclusion and future work

In this paper, we described exploiting a supervised machine learning technique to develop a novel example-based transfer tool that automatically induce Arabic-to-English transfer rules from chunks of relatively small aligned parallel linguistic resources. This tool is very important for those who would like to do Arabic machine translation research but find that the parallel linguistic resources for their translation task are neither available nor affordable. Low density Arabic Script languages like Pashto/Farsi/Amharic can also benefit from the rule induction approach in building their transfer-based machine translation systems.

During the course of rule induction a morphological analysis process is needed, which entails the acquisition of linguistic knowledge. Again, the role of supervised machine learning technique comes to play in order to develop a novel morphological acquisition tool that incrementally acquires induced morphological rules. The rules are induced from a set of monolingual example pairs of inflected forms and their stems based on their vocal patterns and feature structure. This tool is very important for a morphologically rich language like Arabic as it automates the acquisition of morphological rules from examples rather than relying on hand-crafted linguistic rules acquired from Arabic specialists.

This research discusses how to build from a relatively small aligned parallel data set a machine translation system that can make use of induced Arabic-to-English syntactic transfer rules. Although much work has shown that large amount of data can successfully be used to build a machine translation system, we still need to prove that with small amounts of data we can also successfully build a machine translation system. For example, domain-specific machine translation systems usually do not require large amount of data to get acceptable translations.

To demonstrate the capability of this automated technique we conducted rule-based translation experiments that use induced rules from a relatively small dataset. The rule induction experiment achieved a negative exponential growth of the newly induced rules by increasing the number of the training examples which leads to saturation of the transfer rules after a specific number of examples. The translation quality experiment achieved good results in terms of both coverage and quality.

Much remains to be done in the field of rule induction in the context of machine translation. One possible future direction is to reduce the tradeoff between coverage and quality of translation rules. Another direction is to investigate how to apply statistical weights on the induce rules to resolve possible ambiguities.

References

Abdel Monem, Azza, Shaalan, Khaled, Rafea, Ahmed & Baraka, Hoda. 2008. Generating Arabic Text in Multilingual Speech-to-Speech Machine Translation Framework, *Machine Translation*, Springer, Netherlands, 20(4): 205-258.

Ambati, Vamshi & Rohini, U. 2007. A hybrid approach to example based machine translation for Indian languages. *ICON-2007: 5th International Conference on Natural Language Processing*, IIIT Hyderabad, India.

Aramaki, Eiji, Kurohashi, Sadao, Kashioka, Hideki & Kato, Naoto. 2005. Probabilistic model for example-based machine translation. *In Proceedings of MT Summit X*, 219–226. Phuket, Thailand

Beesley, Kenneth. 1996. Arabic finite state morphological analysis and generation. *In Proceedings of the 16th conference on computational linguistics*, 89 - 94

Carl, Michael, Pease, Cartherine, Iomdin Leonid & Streiter, Oliver. 1998. Towards dynamic linkage of Example Based and Rule-Based Machine Translation. *In proceedings of ESSLLI '98 Machine Translation Workshop*.

Chen, Yu, Eisele, Andreas, Federmann Christian, Hasler, Eva, Jellinghaus Michael, Theison Silke. 2007. Multi-engine machine translation with an open-source decoder for statistical machine translation. *In proceedings of Association of Computational Linguistics (ACL 2007).*

Dans, Koriche. 2005. Online closure based learning of relational theories. *In ILP'05: Inductive Logic Programming, Bonn*, Germany, 172-189.

Eineborg, Martin & Lindberg, Nikolaj. 1999. *ILP in part of speech tagging - an overview.* Learning Language in Logic: 57 – 169.

Green, T. 1979. The necessity of syntax markers, two experiments with artificial languages. *Journal of verbal learning and behaviour,* 18(4):481–496.

Hossny Ahamed Shaalan, Khaled & and Fahmy, Aly, 2008. Automatic morphological rule induction for Arabic. *In Proceedings of the Workshop on Human Language Translation and Natural Language Processing within the Arabic World (LREC'08)*, 97-101.

Hossny Ahmed, Shaalan, Khaled, Fahmy Aly, 2009. Machine translation model using inductive logic programming. *In Proceedings of IEEE international conference on natural language processing and knowledge engineering (IEEE NLP-KE'09)*, Dalian, China, 103-110.

Imamura, Kenji, Okuma, Hideo, Watanabe, Taro & Sumita, Eiichiro 2004. Example-based Machine Translation Based on Syntactic Transfer with Statistical Models. *In Proceedings of the 20th International Conference on Computational Linguistics,* Geneva*, 99-105*

Koehn, Philipp, Och, Franz & Marcu, Daniel 2003. Statistical phrase-based translation. *In Proceedings of the Joint Human Language Technology Conference and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, Edmonton, Canada, 127–133.

Lavie, Alon, Probst, Katharina, Peterson, Erik, Vogel, Stephan, Levin, Lori, Font-Llitjos, Ariadna & Carbonell, Jaime. 2004. A trainable transfer-based MT approach for languages with limited resources. *In Proceedings of workshop of the European association for machine translation (EAMT-2004)*, Valletta, Malta, 116-123.

Leusch, Gregor, Ueffing, Nicola, & Ney, Hermann. 2006. CDER: efficient MT evaluation using block movements. *In Proceedings of the 11th conference of the European chapter of the association for computational linguistics*, 241–248

Lindberg, Nikolaj & Eineborg, Martin. 1999. Improving part of speech disambiguation rules by adding linguistic knowledge. ILP: 186-197.

Muggleton, Stephen. 1999. Inductive logic programming issues, results and the challenge of learning language in logic. In *Artificial Intelligence*, 114(1-2), 283-296.

Och, Franz & Ney, Hermann. 2000. Improved statistical alignment models. *In Proceeding of the 38th Annual meeting of the association for computational linguistics*, Hongkong, China, October, 440-447.

Probst, Katharina, Levin, Lori, Peterson Erik, Lavie, Alon & Carbonell, Jaime . 2002. Machine translation for minority languages using elicitation based learning of syntactic transfer rules. *Machine Translation*, 17(4):245-270.

Riezler, Stefan & Maxwell, John. 2006. Grammatical machine translation. *In Proceedings of HLT-NAACL '06 Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA*, 248–255.

Shirai, Shirai, Bond, Francis & Takahashi, Yamato. 1997. A hybrid rule and example-based method for machine translation. *In proceedings of the natural language processing Pacific rim symposium*, Phuket, Thailand, 49-54.

Stroppa, Nicolas, Groves, Declan, Way Andy, Sarasola, Kepa, 2006. Example based machine translation of the Basque language. *In Proceedings of the 7th biennial conference of the Association for Machine Translation in the Americas*, pages 232-241, Cambridge, Massachusetts.

Sumita, Eiichiro, Akiba, Yasuhiro, Doi, Takao, Finch Andrew, Imamura, Kenji, Okuma, Hideo, Paul Michael, Shimohata, Mitsuo, Watanabe, Taro, 2004. EBMT, SMT, Hybrid and More: ATR spoken language translation system. *In Proceeding of International Workshop on Spoken Language Transl*ation, Kyoto, Japan, 13–20.

Veale, Tony, Way, Andy. 1997. Gaijin: A bootstrapping, template-driven approach to example-based machine translation. *In Proceedings of the New Methods in Natural Language Processing (NeMNLP97)*, Sofia, Bulgaria, 239-244.