International Journal of Information Science and Management

A Method to Convert Sana'ani Accent to Modern Standard Arabic

G. H. Al-Gaphari, Ph.D.
Department of Computer Science
University of Sana'a, Yemen
Corresponding Author: drghalebh@yahoo.com

M. Al-Yadoumi, Ph.D.
Department of Electrical Engineering
University of Sana'a, Yemen
email: alyadoumi@hotmail.com

Abstract

This paper presents an efficient mechanism to convert Sana'ani dialect to modern standard Arabic. The mechanism is based on morphological rules related to Sana'ani dialect as well as Modern Standard Arabic. Such rules facilitate the dialect conversion to its corresponding MSA. The mechanism tokenizes the input dialect text and divides each token into stem and its affixes; such affixes can be categorized into two categories: dialect affixes and/or MSA affixes. At the same time, the stem could be dialect stem or MSA stem. Therefore, our mechanism, implemented by using a simple MSA stemmer, must pay attention to such situations. Then our dialect stemmer is applied to strip the resulting token and extract dialect affixes. At this point, the rules are applied to decide when to carry out the extraction of an affix. The experiment shows that Sana'ani dialect has three classes of distortions, which are prefixes, suffixes, and stems distortions. The algorithm normalizes such distortion based on the morphological rules. For each morphological rule the mechanism checks possibility of applying such a rule. That means if rule conditions be met, then the dialect affix will be replaced by its corresponding MSA. If there is no restriction on applying the rule related to the distorted stem, then the rule can be considered as a parallel corpus of the dialect and MSA. Finally, the experiment computes the distortion ratio of MSA in Sana'ani dialect. For a Sana'ani dialect sample of 9386 words, 16.29% of them have distorted suffixes, 0.70% have distorted prefixes and 2.17% contain distorted stems. These percentages are related only to the processed words.

Keywords: Dialects, NLP, Algorithms, MSA, Methods, Conversion.

Introduction

Arabic language garners attention in the Natural Language Processing (NLP) community due to its linguistic difference from Indo-European languages. Modern Standard Arabic (MSA) is a form of Arabic language that is used widely in news media and formal speeches (Diab, Hacioglu, & Jurafsky, 2004). There are no native speakers of MSA as stated in Mutahhar and Watson (2002). The importance of processing the dialects comes from here: "Almost no native speakers of Arabic sustain continuous spontaneous production of MSA. Dialects are the primary form of Arabic used in all

unscripted spoken genres: conversations, talk shows, interviews, etc." (Habash & Rambow, 2005). Dialects are increasingly in use in new written media (newsgroups, weblogs, online chat etc). "Substantial Dialect-MSA differences impede direct application of MSA NLP tools" (Diab & Habash, 2006). Converting Sana'ani dialect to MSA enables MSA NLP tools to process this dialect indirectly. For example, it is easy to parse the translated dialect using MSA parser rather than developing a special parser for that dialect (Al-Razi & Elsehah, 1996). In this example, suboptimal output MSA can still be helpful for the parsing task without necessarily being fluent or accurate, since the goal is parsing (Chiang, Habash, & Rambow , 2005). Such fields of researches suffer from lack of resources due to lack of standards for the dialects, as well as lack of written resources of dialects themselves as shown in Maamouri and Bies (2004). Dialect to MSA translation is useful in many other applications, such as querying dialect text using MSA words as discussed in Ferguson (1959). Search engines should retrieve appropriate results from dialect pages as well as MSA ones with the same query that is usually written in MSA as reviewed in Bar (2006).

It is right to say that the dialect is very popular where the majority of people use it during their daily life, they use it for conversation and online chatting as well. Unfortunately, not only is the dialect rarely used in writing, but it also has no written standard. It is realized as a language of heart and feeling where MSA is considered as a language of mind. It is a formal language that has a very good written standard. The majority of educated and rational people keep writing their ideas and documents using MSA. Also the most important electronic documents are available in this standard language. Therefore, it is very important to find an electronic tool to convert the dialect into MSA, which will in turn be processed by existing NLP techniques, rather than building such techniques for processing dialect itself.

There is a limited number of Arabic dialect softwares developed and a limited number of research papers published. Taghva, Elkhoury, and Coombs (2005) have built a software algorithm to analyze Arabic Stemming without a root dictionary. Mutahhar and Watson (2002) present a report on social issues in popular Yemeni culture. The report is written in modern Sana'ani dialect with English translation. However, the report lacks a software algorithm (Mutahhar & Watson, 2002). In general, we can say that a little or no effort has been made regarding the Sana'ani dialect-MSA conversion.

In fact, the main objective of this paper is to design and implement an algorithm to convert the Sana'ani dialect into modern standard Arabic.

Method

The primary goal of this study is to design and implement an efficient method to convert Sana'ani dialect to MSA. The algorithm concentrates on MSA tokenization,

MSA stemming, processing the dialect, translating the dialect and rebuilding the token. That is implemented by using a simple MSA stemmer. Then our dialect stemmer is applied to strip the resulting token and extract dialect affixes. While analyzing the Sana'ani dialect, we found out the majority of distorted words in the dialect depend on the previous word, the next word, and /or their tags. Therefore, the distortion can be solved by understanding the distorted word context. Such understanding could be obtained through implementing some technical tools such as grammatical rules and lexicon for the neighboring words senses, their tags, and their roles in such context. In fact, this section is dictated to build and implement:

Syntactic Rules

In this section we attempt to build some syntactic rules to control the replacement mechanism of the distorted words by its MSA equivalent. These rules could be applied to handle any distortion in MSA language (Sana'ani dialect). They depend on pattern length of two words in addition to the distorted word that falls in focus. The two words may be any of MSA words or two words types as shown in Table 1.

For dialect clitic that has more than one rule, the rules must be arranged from the more specific to less specific. 'Previous' stands for any appearance of word in previous words with respect to the word that has distortion (the word in focus) within the sentence.

Table 1

Syntactic Rules

Rules Name	Rule Type	Previous word	Previous word type	Next word	Next word type	MSA equivalent
ـش	Enclitic	ما	Null	Null	Null	Empty string
	Enclitic	Null	Null	Null	Null	كِ (feminine, 2nd person)
عـ	Proclitic	Null	Null	Null	Verb,Muthare3	سـ
مش	Stem	Null	Null	Null	Null	ليس، ما

The translation process from dialect to MSA could take place as shown in Table 2

Table 2

Sample of Translation

No#	Example #1	Example #2	Example #3
Sana'ani dialect	ما قدرتـ**ش** العب معهم mA qdrt_**\$** AlEb mEhm	لعبـ**ش** **مش** هو حالي lEb_**\$** **m\$** hw HAly	ما عـ**تقدرش** تلعب معهم mA **E**__tqdr**\$** tlEb mEhm

MSA: Distortion correction	ما قدرت ألعب معهم mA qdrt >lEb mEhm	لعبكِ ما هو حالي lEbki mA hw HAly	ما ستقدر تلعب معهم mA stqdr tlEb mEhm
MSA: Grammar and fluency consideration	ما أستطعت أن العب (اللعب) معم mA >stTEt >n AlEb (AllEb) mEm	لعبكِ ليس حسن lEbki lys Hsn	لن تقدر(تستطيع) ان تلعب معهم ln tqdr(tstTyE) An tlEb mEhm
Distortion type	Enclitic distortion at '**ـش**' in '**قدرتـش**'	Enclitic distortion at '**ـش**' in '**لعبـش**' and stem distortion in 'مش'	Proclitic distortion at '**عـ**' and enclitic distortion at '**ش**' in '**عتقدرش**'

In example 1 and example 2 included in Table 2, different rules are selected and applied to the same enclitic 'ش'. If the rules are not prioritized, there is no guarantee to apply 'empty string' rule of 'ش' at all, because the other will be applicable all the time. To apply any rule we should use a stemmer to divide a token into smaller ones. This stemmer should not be a complex stemmer, but a suitable one, that is capable to stem a token in parts which are understandable by rules. Such a stemmer is sufficient for that task.

Stemming Process

Before applying the rule, we must provide that with an outcome of stemmer. This is important because the Sana'ani dialect contains MSA clitics (as shown in Table 3) or dialect clitics with MSA word or a combination of both. Each one of them must be divided apart to be easily processed by rules (Habash, Rambow, & Kiraz, 2005). We borrowed our stemmer idea from ISRI stemmer which is an automatic stemmer without root dictionary (Taghva et al., 2005).

Rules that need to specify word types can be solved by using an automatic classification of words in Arabic language or by using a dictionary containing MSA stems along with their types. A dictionary of words with the types mentioned in the rules is sufficient to conduct the task.

Table 3

MSA Clitics Sets

Set	Description	Examples
D	Diacritics- owelizations	'ً','ٌ','ٍ','َ','ُ','ِ','ّ','ْ'
P3	Prefixes of length three	"بال", "كال", "وال", "ولل"
P2	Prefixes of length two	"ال", "لل"
S3	suffixes of length three	"تما", "هما", "تان", "تين", "كما"
S2	suffixes of length two	"ين","ان","ات","ون","نا","هن","كم","تن","كن","تم","ها","يا",

Algorithm steps

The Proposed method given in the previous section can be written in the form of an algorithm to be implemented, the steps of the algorithm are as follows:

1. Remove diacritics representing vowels. Set D on Table 3.
2. Remove connector و if it precedes a word beginning with و.
3. Remove length three denoted by S3 and length two denoted by S2 which are suffixes of MSA in that order. Extract them from the token. MSA suffixes are shown in Table 3
4. Remove length three denoted by P3 and length two denoted by P2 which are prefixes of MSA in that order. Extract them from the token. MSA prefixes are shown in Table 3 as well.
5. Replace suffixes of dialect with their MSA alternatives according to the rules whenever their conditions are met. Process the longer prefixes first then shorter ones. Extract them from the token.
6. Do the same (step 5) with prefixes of dialect.
7. Extracting stem by removing dialect clitics regardless of applicable of rule may resolve the dependency of rule. Rule may not be applied until next word is processed. Next word also may need processing of the previous one. In general, our rules do not have such a deep dependency except in distorted MSA stems which have dialect clitics.
8. Check the remaining token. If it is a dialect stem, apply stem rules and get the alternative MSA stem.
9. Rebuild the token by adding the removed MSA clitics and dialect repaired clitics to the stem.
10. Unrecognized tokens remain unchanged.

The Previous steps have been converted to a flowchart as shown in Figure 1.

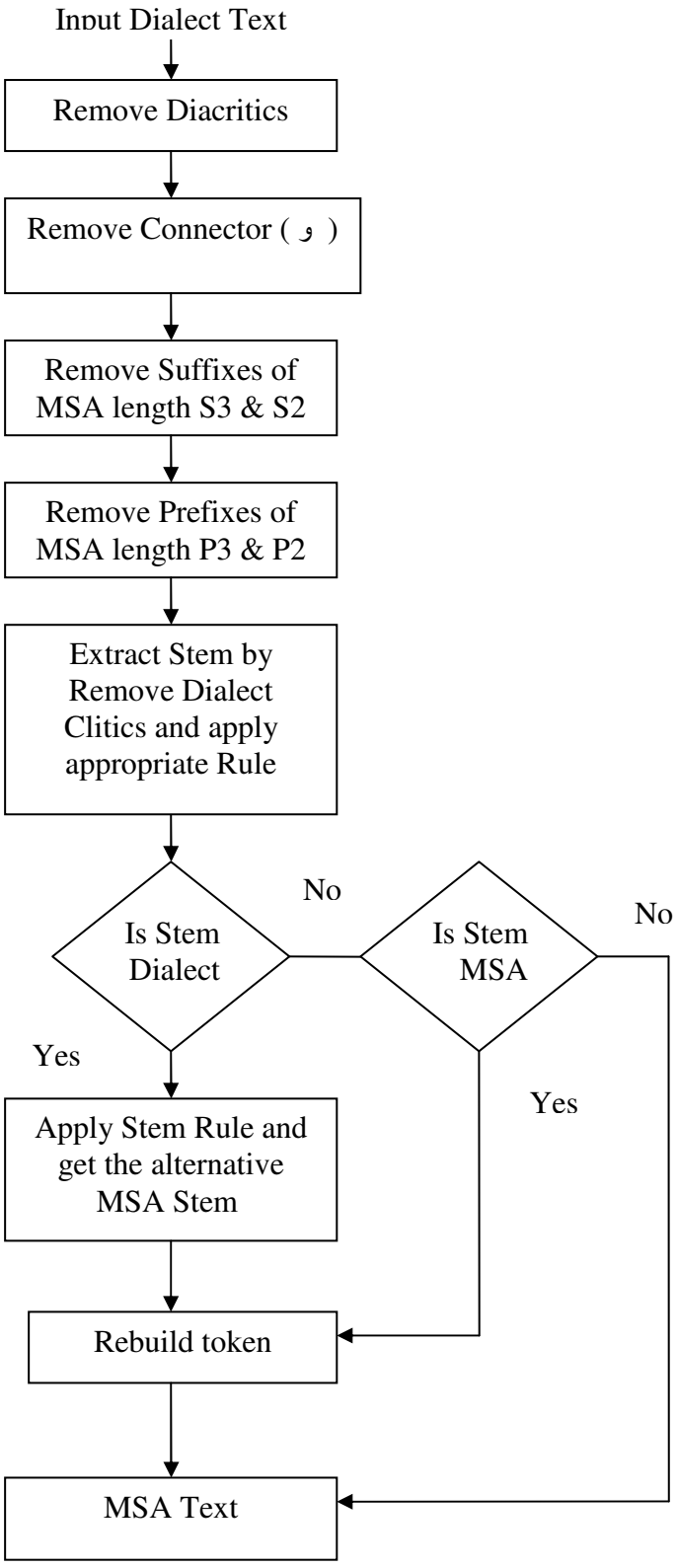

Figure 1. Conversion algorithm.

Results

We have applied the algorithm on a sample of Sana'ani dialects. The algorithm accepts Sana'ani dialect text as inputs, it processes the corpus and produces Table 4 contents. The size of the corpus content is about 80 kilo bytes. It is a textual comedy program produced by Sana'a Broad Casting. It describes many social, economical, educational, and cultural issues using the Sana'ni dialect.

The sample size was limited because of the lack of electronic San'ani dialect text availability. Fortunately, the sample corpus selection was appropriate to reflect the reality of Sana'ani accent. It could be a very good selection criterion to reflect the status quo problem. The corpus is processed automatically in NLP algorithms, where an object oriented program is written in C# for the corpus processing. The corpus is automatically read and tokenized, hence terms' frequencies are computed. Finally, stemming and conversion took place based on morphological and syntactic rules that were stored in a lexicon. The experiment process could be shown as fallows:

Lexicons

Two lexicons were used: the first one to hold rules and the other to hold corpus terms and their features. Such lexicons are implanted using C# ArrayList and Hashtable within a class that includes manipulation methods such as "Addrule" and "Addword".

Application data set

The serialization mechanism was used to store the application lexicons on the hard disk.

```
Public Dict (String file Name)
{
    If (! File. Exists (file Name))
    {
        Table= new Hashtable();
        Return;
    }
Stream myfilestream= File . OpenRead(filename)
IDictionary dictTable=(hashtable)(deserializer. Deserialize(myFileStream));
MyFileStream.(losec).
}
```

Then the Hashtable is saved into the file in the same manner by another method called SaveToFile( ).

Steady Design

A dynamic link library was developed for any multithreaded application that works without any conflict or losing of integrity. It is implemented using **lock ()** technique.

IComparable Interface

Matching of applicable rules on dialectal sentences starts with the longer rule name first to make the stemming process much safer. Then sorting rules within the ArrayList was set up as an enabled.

Multithreading

The application user interaction was implemented by using Application.DoEvents() method that makes the application robust and efficient in terms of concurrent execution.

Inherited Form

The application includes multiple document interface (MDI) type. It has a child **form** for the dialect text and another one for the MSA text. They have the same properties and methods. A dialect child form was developed and later the other form was inherited.

As a result of the above processing, the corpus is analyzed into 9386 words; such results are restricted to our application and related only to the processed words. The results are governed by the total data on the dialect-MSA differences, stored in the application. It, for example, includes stored rules to the all the possible rules, and the size of parallel corpus as well as stored tagged words.

Table 4

Experiment Results

Affix	Ratio
Processed MSA suffixes	843
Processed MSA prefixes	1570
Processed MSA stems	1014
Processed dialect suffixes	164
Processed dialect prefixes	11
Processed dialect stems	204
Distortion of MSA suffixes	14.286%
Distortion of MSA prefixes	0.696%
Distortion of MSA stem	2.17%
Total words	9386

Discussion

The algorithm is capable of converting 77.32 % of the whole corpus from Sana'ani dialect into MSA. That means the algorithm works fine as long as it is able to accept Sana'ani dialect of size 9386 words and process such a corpus to produce 77.32 % of that corpus as a normal MSA. The other part of the corpus represents 22.68% of the whole corpus, such part includes distorted and ambiguous words.

The experiment shows that the distortion rate of MSA in Sana'ani dialect is not too high. The total distortion represents 17.2% of the whole corpus. Most of that distortion occurs in MSA suffixes that represents 14.286% of the total corpus. While the distortion of MSA stems and MSA prefixes represent 2.17% and 0.696 % of the total corpus, respectively. Also, the experiment shows the ambiguous words represent 5.48% of the whole corpus. Such words are non-translated words. The ambiguity is a well known problem; it occurs in different situations and requires different tools and techniques to be handled. In fact, the ambiguous results in this experiment occur as a result of mapping one dialect stem and/or rule to two or more equivalents of MSA. For example, Sana'ani dialect word ' لا – *lA*' has more than one meaning in MSA. One of the meanings (table 5-a) is "to", the other meaning is "no" or negation (table 5-b), while the third one is "if" as in table 5-c. this is a new type of ambiguity, the same pronunciation with different meanings. Such ambiguous situations require an inference system or an machine understanding to select the coherent one.

Table 5a

The First Meaning of (لا- lA) in MSA: To

Sana'ani dialect translit	أخي عند **لا** اسرح Axy End **lA** AsrH
MSA gloss	أخي عند **إلى** اذهب brother round **to** go
Meaning	Go to my brother

Table 5b

The Second Meaning of (لا- lA) in MSA: Negation

Sana'ani dialect translit	شي **ولا** ابنش مع تتعصبي **لا** \$y w**lA** Abn\$ mE ttESby **lA**
MSA gloss	شيء **ولا** ابنكِ مع تتعصبي **لا** thing and **no** your son with gang up **don't**
Meaning	You should **not** gang up with your son like that!

Table 5c

The Third Meaning of (لا- lA) in MSA: If

Sana'ani dialect	... اسرح ، تكسب تشتي انت **لا**
Translit	... AsrH , tksb t\$ty Ant **lA**
MSA	... اسرح، تكسب تريد أنت **إذا**
Gloss	… go win want you **if**
meaning	If you want to win, go …

There is another problem related to the approach itself. That is, a rule may need to know neighbors of in-process token, while processing the neighbors may also need processing the token first. This is called "the dependant rules problem". We should use effective technique to solve this problem. We can partially cope with it, using the original sentence while processing each token. The other dependant rule will be applicable by removing clitics (stemming) and/or replacing dialect stem with MSA equivalent using corpus.

Conclusion

The experiment results show how to use rule-based algorithm to convert the dialect to MSA. The algorithm is able to convert 77.32 % of the whole corpus from Sana'a ni dialect into MSA. That means the algorithm works fine as long as it is able to obtain such a percentage. It, also, shows the total distortion represents 17.2% of the whole corpus. Most of that distortion occurs in MSA suffixes that represents 14.286% of the total corpus. While the distortion of MSA stem and MSA prefixes represent 2.17% and 0.696 % of the corpus, respectively. Moreover, the experiment results show the ambiguous words represent 5.48% of the whole corpus.

The accuracy could be increased by addressing some weaknesses in the implementation. This work can be combined with other dialects processing techniques to build an effective tool that is capable of translating dialect to MSA, with high accuracy. Rules tags can be expanded and enhanced to include extra fields in order to remove uncertainty in applying the rules. Problem of inputting the tags manually can be solved by developing techniques similar to automatic tagging used in Arabic. MSA-Dialect stem corpus is still important, so large effort should be made to build it and tags should be included in the corpus for each stem.

References

Al-Razi, M. & El-Sehah, M. (1996). *An Arabic-Arabic root dictionary*. Beirut: Al-Resalah Publishing House.

Bar, K. (2006). Machine translation. *NLP Conference*. Retrieved from http://www.

cs.tau.ac.il/~ nachumd/ NLP/MT.pdf

Chiang, D., Diab, M., Habash, N., Rambow, O., & Shareef, S. (2005). Parsing Arabic dialects. *JHU Summer Workshop*. Final Report.

Diab, M. & Habash, N (2006). *Arabic dialect processing*. Bostan: AMTA.

Diab, M., Hacioglu, K., & Jurafsky, D. (2004). Automatic tagging of Arabic text: From row text to base phrase chunks. *In 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLTNAACL04),* Boston, MA.

Ferguson, C. F. (1959). Diglossia. *Word, 15* (2), 325–340.

Habash, N. & Rambow, O. (2005). Tokenization, morphological analysis, and part-of-speech tagging for Arabic in one fell swoop. In *Proceeding of the Association for Computational Linguistic (ACL).*

Habash, N., Rambow, O., & Kiraz, G. (2005). *Morphological analysis and generation for Arabic dialects*. Center for Computational Learning Systems. Columbia University.

Maamouri, M. & Bies, A. (2004). *Developing an Arabic treebank: Methods, guidelines, procedures, and tools.* Linguistic Data Consortium (LDC).

Mutahhar, A. R. & Watson, J. (2002). *Social issues in popular yemeni culture.* Yemeni-British project supported by the British Embassy, Socianl Fund for Development and Leigh Douglas Memorial Fund, Sana'a, Yemen.

Taghva, K., Elkhoury, R., & Coombs, J. (2005). *Arabic stemming without a root dictionary*. Information Science Research Institute (ISRI), Las Vegas, University of Nevada.