2012 AMTA 20 Years

The Tenth Biennial Conference of the
Association for Machine Translation in the Americas

Computational Approaches to Arabic Script-based Languages

Ali Farghaly

Farhard Oroumchian

University of Wollongong in Dubai,
United Arab Emirates

SAN DIEGO, CA
OCTOBER 28- NOVEMBER 1, 2012

The Fourth Workshop on Computational Approaches to Arabic Script-based Languages Proceedings

AMTA 2012 - San Diego, CA , USA

November 1, 2012

ALI FARGHALY

FARHAD OROUMCHIAN (Eds.)

Preface

This is the fourth of a series of workshops designed to bring together researchers working in all languages that use the Arabic script. The absence of short vowels and other diacritic marks from the Arabic script greatly compounds the ambiguity problem which challenges NLP applications. The historical interaction between the Arabic language and culture on the one hand and the other languages and cultures that adopted the Arabic script created a lasting commonality among all Arabic script-based languages. For example a named entity recognition system in any Arabic script-based language has to deal with the problem of the lack of capitalization, the absence of short vowels, and the lack of the strict format of names that is usually observed in Western names. For example, concepts such as last names, given names, maiden names, other name are not often adhered to in names of people in countries that use the Arabic scripts. This workshop dedicates a whole session to the discussion of name matching and named entity recognition.

Since this workshop is hosted by AMTA 2012, it is not surprising to see more than a third of the accepted papers deal directly with issues Arabic and Farsi machine translation. These papers deal with challenging problems in machine translation such as the translation of idiomatic and multi word expressions, the problem of translating discourse connectives and the issues encountered in the design of open domain machine translation systems for Farsi.

We look forward to our fifth workshop which we hope we have more papers on languages other than Arabic and more work that compares challenges and solutions in one task across different Arabic script-based languages.

November 2012 Ali Farghaly

ORGANIZATION

ORGANIZING COMMITTEE

ALI FARGHALY

STANFORD UNIVERSITY AND MONTEREY PENINSULA COLLEGE, USA

FARHAD OROUMCHIAN

UNIVERSITY OF WOLLONGONG AT DUBAI, UNITED ARAB EMIRATES

INVITED SPEAKER

HASSAN SAWAF

CHIEF SCIENTIST, SAIC, USA

TITLE OF THE TALK:

More Than 20 Years of Machine Translation of Arabic-Script Languages: Overview of the History of Diverse Challenges in Research and Deployment

PROGRAM COMMITTEE

TIM BUCKWALTER	UNIVERSITY OF MARYLAND, USA
VIOLETTA CAVALLI-SFORZA	AL AKHAWAYN UNIVERSITY, MOROCCOA
SHERRI CONDON	MITRE, USA
MONA DIAB	COLUMBIA UNIVERSITY, USA
SARMAD HUSSAIN	CRULP, PAKISTAN
FARHAD OROUMCHIAN	UNIVERSITY OF WOLLONGONG IN DUBAI, UAE
KHALED SHAALAN	THE BRITISH UNIVERSITY IN DUBAI,UAE
AHMED RAFEA	THE AMERICAN UNIVERSITY IN CAIRO, EGYPT
IMED ZITOUNI	IBM, USA

WORKSHOP PROGRAM

OPENING SESSION

9:00 – 9:30 ***Ali Farghaly, Organizer***

Commonalities in Arabic Script-based Languages: An Example from Name Matching

SESSION 1

9:30 – 10:30 ***Hassan Sawaf, Invited Speaker***

More than 20 years of Machine Translation of Arabic-Script Languages:

Overview of the History of Diverse Challenges in Research and Deployment"

10:30 – 11:00 BREAK

SESSION 2 MACHINE TRANSLATION

11:00 – 11:30 Translating English Discourse Connectives into Arabic: a Corpus-based Analysis and an Evaluation Metric

Najeh Hajlaoui and Andrei Popescu-Belis
Idiap Research Institute

11:30 – 12:00 Idiomatic MWEs and Machine Translation. A Retrieval and Representation Model: the AraMWE Project

Giuliano Lancioni and Marco Boella
Roma Tre University, Italy, 2Rome University "La Sapienza", Italy

12:00 - 12:30 Developing an Open-domain English-Farsi Translation System Using AFEC: Amirkabir Bilingual Farsi-English Corpus

Fattaneh Jabbari, Somayeh Bakhshaei, Seyed Mohammad Mohammadzadeh Ziabary and Shahram Khadivi
Amirkabir University of Technology, Terhan, Iran

12:30 – 2:00 LUNCH

SESSION 3 ENTITY RECOGNITION

2:00 – 2:30 **ARNE - A tool for Named Entity Recognition from Arabic text**

Carolin Shihadeh and Günter Neumann

DFKI, Saarbr¨ucken, Germany

2:30 – 3:00 **Approaches to Arabic Name Transliteration and Matching in a Software Knowledge Base**

Brant Kay and Brian Rineer
SAS Institute Inc., Cary, North Carolina, USA

3:00 – 3:30 **Using Arabic transliteration to improve word alignment from French-Arabic parallel corpora**

Houda Saadane, Nasredine Semmar, Ouafa Benterki and Christian Fluhr
LIDILEM - Université Stendhal Grenoble 3, Cedex, France
Institut Supérieur Arabe de Traduction, Bir Mourad Raïs, Algérie

3:30 – 4:00 BREAK

SESSION 4 SENTIMENTS AND MORPHOLOGICAL TAGGING

4:00 – 4:30 **Preprocessing Egyptian Dialect Tweets for Sentiment Mining**

Amira Shoukry and Ahmed Rafea
The American University in Cairo, Cairo, Eygpt

4:30– 5:00 **Rescoring N-Best Hypotheses for Arabic Speech Recognition: A Syntax-Mining Approach**

Dia Eddin AbuZeina, Moustafa Elshafei, Husni Al-Muhtaseb and Wasfi Al-Khatib
Palestine Polytechnic University, Hebron, Palestine
King Fahd University of Petroleum and Minerals, Saudi Arabia

5:00 - 5:30 **Morphological Segmentation and Part of Speech Tagging for Religious Arabic**

Emad Mohamed
Carnegie Mellon University Qatar

Table of Contents

Translating English Discourse Connectives into Arabic: a Corpus-based Analysis and an Evaluation Metric

Najeh Hajlaoui
Idiap Research Institute
Martigny, Switzerland
najeh.hajlaoui@idiap.ch

Andrei Popescu-Belis
Idiap Research Institute
Martigny, Switzerland
andrei.popescu-belis@idiap.ch

Abstract

Discourse connectives can often signal multiple discourse relations, depending on their context. The automatic identification of the Arabic translations of seven English discourse connectives shows how these connectives are differently translated depending on their actual senses. Automatic labelling of English source connectives can help a machine translation system to translate them more correctly. The corpus-based analysis of Arabic translations also enables the definition of a connective-specific evaluation metric for machine translation, which is here validated by human judges on sample English/Arabic translation data.

1 Introduction

Discourse connectives are a class of lexical items which signal discourse relations between clauses or sentences. Several discourse connectives that are frequent in English are also quite ambiguous, in that, depending on their occurrence, they can signal various discourse relations. When translating from English into another language, this ambiguity can lead to wrong translations, if the target connective conveys an unintended discourse relation. For instance, *since* can have a causal or a temporal sense, and, depending on the target language, these senses can be translated by different connectives. In other cases, a connective may be translated by a different construction (reformulation) or even be skipped in translation.

We consider here seven frequent English discourse connectives: *although, even though, meanwhile, since, though, while,* and *yet*. Previous studies have shown that it is possible to disambiguate their main senses automatically with acceptable accuracy (Pitler and Nenkova 2009), and that the sense labels can be used by machine translation (MT) systems to improve their translation (Meyer and Popescu-Belis 2012). For instance, when translating from English to French, a statistical MT (SMT) system can use parallel corpora with labelled connectives to learn correct translations based on labels. One issue with such experiments is the capacity to measure the translation improvement due to the correct translation of connectives, for instance by focussing only on these lexical items.

In this paper, we explore the translation of the seven above-mentioned English discourse connectives into Arabic. We study to what extent the ambiguities of these connectives are reduced (or not) by translation into Arabic, i.e. if different senses are always translated by different Arabic connectives. Indeed, while a corpus with sense-annotated Arabic discourse connectives has been announced (Al-Saif and Markert, 2010), little has been published about their possible senses. Our analysis is a contribution towards the construction of a full dictionary of Arabic discourse connectives listing their possible senses with observed

frequencies.

This paper has also a second, more pragmatic goal. Our corpus-based analysis was used to define a dictionary of acceptable *vs*. unacceptable "synonyms" for Arabic discourse connectives, which is used for automating the evaluation of English/Arabic MT with respect to connectives. We thus define and assess (meta-evaluate) an automatic metric that estimates how many connectives are correctly translated. The metric (called ACT for Accuracy of Connective Translation) is similar in concept to a BLEU or METEOR metric restricted to discourse connectives, and is shown to have about 90% accuracy.

The paper is organized as follows. In Section 2, we present the empirical study of Arabic translations of English discourse connectives. In Section 3 we present the principle of the ACT metric, and in Section 4 we give meta-evaluation results, along with sample results from a baseline English/Arabic SMT system.

2 Translations of English Discourse Connectives into Arabic

2.1 Ambiguity of Discourse Connectives

The manual annotation of discourse relations in the Penn Discourse Treebank (PDTB) (Prasad et al., 2008) has provided a discourse-layer annotation over the Wall Street Journal Corpus. The annotation targeted either explicit discourse relations (18,459 connectives) or implicit ones (16,053 relations). The sense labels started from top-level senses (temporal, contingency, comparison, and expansion), with 16 subtypes on the second level and 23 subsubtypes on the third level.

In (Al-Saif and Markert, 2010) a manual annotation of Arabic discourse connective has been performed and should be soon available. However, the published material is not explicit about the observed level of ambiguity of Arabic discourse connectives. Rather, the Arabic discourse connectives are only given unique English glosses (implying a 1-to-1 relation), but as we show below for *although* or *since*, the translation is rather a 1-to-n relation.

Discourse connectives can indeed signal several types of discourse relations; the meaning of an occurrence thus varies depending on the context. For example, the English connective '*since*' can have two senses:

- a causal sense which can be translated to Arabic by "*nZrA* نظرا", "*b+ AlnZr* بالنظر", "*AEtbArA* اعتبارا", etc.
- a temporal sense which can be translated to Arabic by "*mn** منذ", "*m** مذ", "*TAlmA* طالما", etc.

Other English connectives can express concession and contrast relations. The English connective *although*, for example, can express a contrast relation, which can be translated to Arabic by "*gyr An* غيرأن", or by "*lkn* لكن", but can also convey a concessive meaning which can be translated in Arabic by "*Alrgm*" الرغم", or "*rgm* رغم". As the translation of an English connective to Arabic varies depending on the intended discourse relation, an MT system that is capable to modulate the translation accordingly should avoid mistakes observed with current systems. Consequently, the MT evaluation should also take into account the acceptable senses of the connectives.

2.2 Approach and Data

We focus on seven English discourse connectives (*although, though, even though, while, meanwhile, since, yet*), with the goal of finding their correspondences in Modern Standard Arabic (MSA) along with information about translation preferences. Of course, the Arabic translations are not necessarily expected to render specifically each sense of the English discourse connectives, as Arabic connectives may have their own ambiguities. For example, the frequent connective "w و" has six rhetorical types, which can be divided into two classes: segment (*fasl*) and non-segment (*wasl*), see (Iraky et al., 2011). Nevertheless, by looking at possible overlaps between the Arabic translations of the seven English connectives, we also gain information about the ambiguity of Arabic connectives.

In order to find the possible translations of the seven ambiguous English connectives, we used an automatic method based on alignment between sentences at the word level using GIZA++ (Och and Ney, 2000). We experimented with the large UN parallel corpus to find out the Arabic connectives that are aligned to English ones, a corpus of journal articles and news:

- English: 1.2 GB of data, with 7.1 million

sentences and 182 million words.
- Arabic: 1.7 GB of data, with 7.1 million of sentences and 154 million words.

For the alignment task, the data was pre-processed as follows:
- English: tokenisation and lowercase.
- Arabic: word transliteration, and segmentation using MADA (Habash and Rambow, 2005).

2.3 Statistics for Connective Dictionaries

Using the automatic alignment method described above, we extracted the word alignment on the Arabic side given the English one. The following tables (Table 1 to Table 7) show the correspondences between each English connective and Arabic translations detected automatically using the projection from English sentences to Arabic ones.

Because word alignment is not perfect, we observe that the result is not always an Arabic connective, though it generally includes one. The main observation is that the obtained vocabulary is limited around more or less the same terms, which form a limited set of translations for each English connective.

Arabic translations of *although*			
Buckwalter	**Arabic**	**N. of occ.**	**% of total**
Alrgm	الرغم	7,091	20.3%
w+	و	5,634	16.1%
rgm	رغم	5,408	15.5%
w+ Alrgm	والرغم	5,308	15.2%
w+ rgm	ورغم	5,298	15.2%
w+ mE	ومع	2,147	6.1%
mE	مع	1,323	3.8%
w+ kAnt	وكانت	542	1.5%
kAnt	كانت	406	1.2%
w+ lw	ولو	242	0.7%
Others		*1561*	*4.4%*
Total		**34,960**	**100%**

Table 1: Translations of the 34,960 occurrences of *although* with explicit alignments (out of 38,476).

Table 1 shows the Arabic translations of the English connective *although* determined by word alignment. The main correspondences are "rgm رغم", "mE مع", "kAnt كانت", "lw لو". The others correspondences, which represent a very small proportion of the total, also include some of these main words, due to alignment inaccuracies.

Arabic translations of *even though*			
Buckwalter	**Arabic**	**N.**	**%**
w+ Alrgm An	و الرغم ان	296	13.2%
Hty w+ An	حتي و ان	244	10.9%
w+ rgm An	ورغم ان	208	9.3%
mE An	مع ان	167	7.4%
w+ mE An	و مع ان	165	7.4%
w+ An	وان	152	6.8%
w+ Alrgm	والرغم	123	5.5%
Hty w+ An kAn	حتي وان كان	108	4.8%
Hty w+ An kAnt	حتي وان كانت	92	4.1%
w+ An kAn	وان كان	82	3.7%
w+ An kAnt	وان كانت	80	3.6%
w+ rgm	ورغم	69	3.1%
Others		*459*	*20.5%*
Total		2,245	100%

Table 2: Translations of the 2,245 occ. of *even though* with explicit alignments (out of 4,751).

Arabic translations of *though*			
Buckwalter	**Arabic**	**N. of occ.**	**%**
rgm An	رغم ان	330	22.7%
w+ An	وان	274	18.8%
Alrgm An	الرغم ان	235	16.2%
mE An	مع ان	110	7.6%
w+ Alrgm	والرغم	97	6.7%
w+ rgm	ورغم	65	4.5%
Alrgm	الرغم	56	3.9%
rgm	رغم	51	3.5%
w+ Alrgm An	و الرغم ان	47	3.2%
Others		*189*	*11.6%*
Total		**1,454**	**100%**

Table 3: Translations of the 1,454 occurrences of *though* with explicit alignments (out of 3,006).

Arabic translations of *since*			
Buckwalter	**Arabic**	**N. of occ.**	**%**
mn*	منذ	11,165	77.946%
nZrA	نظرا	923	6.444%
Hyv	حيث	851	5.941%
w+	و	543	3.791%
A*	اذ	256	1.787%
[mn*]	[منذ]	179	1.250%
AlnZr	النظر	150	1.047%
Others		*257*	*1.8%*
Total		**14,324**	**100%**

Table 4: Translations of the 14,324 occurrences of *since* with explicit alignments (out of 20,163).

Arabic translations of *yet*			
Buckwalter	**Arabic**	**N. of occ.**	**%**
w+ mE *lk	ومع ذلك	226	22.7%
mE *lk	مع ذلك	182	18.8%
mE *lk f+	مع ذلك ف	133	13.4%
w+ lkn	ولكن	86	8.6%
myyh	مييه	60	6.0%
gyr	غير	52	5.2%
lkn	لكن	34	3.4%
mE	مع	25	2.5%
AlA	الا	25	2.5%
w+	و	24	2.4%
mE h*A f+	مع هذا ف	15	1.5%
*lk	ذلك	14	1.4%
f+	ف	10	1.0%
Others		*110*	*11.030%*
Total		**996**	**100%**

Table 5: Translations of the 996 occurrences of *yet* with explicit alignments (out of 7,087).

We had poor alignment results for *yet* because only 996 occurrences were aligned out of 7087. Consequently, we examined directly all the sentences to find out all the possible translations into Arabic of the English connective *yet*.

Arabic translations of *meanwhile*			
Buckwalter	**Arabic**	**N.**	**%**
w+ Alwqt nfs	والوقت نفس	432	47.0%
w+ Alwqt *At	والوقت ذات	212	23.0%
w+ nfs Alwqt	ونفس الوقت	138	15.0%
w+ gDwn *lk	و غضون ذلك	32	3.5%
Alwqt nfs	الوقت نفس	30	3.3%
Alwqt *At	الوقت ذات	17	1.8%
w+ *At Alwqt	وذات الوقت	15	1.6%
Others		*44*	*4.8%*
Total		**920**	**100%**

Table 6: Translations of the 920 occurrences of *meanwhile* with explicit alignments (of 2,795).

From these tables, it is possible to assign sense labels to the Arabic translations, and therefore perform sense-labeling over the English source connectives, following a "translation spotting" approach as in (Meyer et al. 2011). However, our goal with respect to the evaluation metric is slightly different: we need, for each English source connective, to cluster the possible translations according to their senses, in order to obtain lists of Arabic "synonyms" of discourse connectives.

Arabic translations of 'while'			
Buckwalter	**Arabic**	**N.**	**%**
bynmA	بينما	139	36.0%
w+	و	110	28.5%
Hyn	حين	66	17.1%
mE	مع	54	14.0%
w+ bynmA	وبينما	6	1.6%
w+ mE	ومع	5	1.3%
w+ Hyn	وحين	5	1.3%
tHqyq *At qymp	تحقيق ذات قيمة	1	0.3%
Total		**386**	**100%**

Table 7: Translations of the 386 occurrences of 'while' with explicit alignments (out of 1,002).

2.4 Dictionaries of Connectives

Starting from the above tables, we first cleaned the Arabic vocabulary by merging several translations into one entry. Second, we added other possible (known) translations to complete the dictionary. Third, we classified them by checking the sentences containing these connectives to confirm the exact sense of each connective.

For instance, the possible Arabic translation of "*since"* can be classified along two senses, Temporal and Causal, without any overlap between the two lists, as follows. For lack of space, we list below only the most frequent Arabic translation, and we give only "*although"* because *"though"* and "*even though"* follow the same pattern.

```
$althoughCONTRAST="lw لو|gyr An غير ان|lkn لكن|
lAn لئن|An lm إن لم";

$althoughCONCESSION="Alrgm الرغم|rgm رغم|mE
مع|A*A kAn إذا كان|An kAn إن كان|fy Hyn في حين|
kmA kAn كما كان|AnmA إنما";

$sinceTEMPORAL="mn* منذ| m* مذ|bEd بعد|TAlmA
طالما|mA dAm مادام|wmn*}* منذئذ";

$sinceCAUSAL="nZrA نظرا|b+ AlnZr بالنظر|mE
اعتبارا|AEtbArA لما|lmA اذ|A* حيث|Hyv مع النظر
AlnZr
|b+ mA An بما أن |mA An ما أن|A*A إذا|lAn لأن ";


$yetCONCESSION="mE *lk مع ذلك|mE h*A مع هذا |
mE مع|Ely An على أن";
```

$**yetCONTRAST**="lkn لكن|gyr An غير أن|AlA An إلا أن|byd An بيد أن";

$**yetADVERB**="bEd بعد|lA yzAl لا يزال|Hty AlAn حتى الآن|mA zAl ما زال";

$**whileCONTRAST**="mE An مع أن|mE مع|lAn لأن|lkn لكن";

$**whileCONCESSION**="Alrgm الرغم|rgm رغم|A*A إذا|A* اذ";

$**whileTEMPORAL**="bynmA بينما|Ely Hyn على حين|fy Hyn في حين";

3 Evaluation of Connective Translation

3.1 ACT Metric

Distance-based MT evaluation metrics compute a distance between the MT output (candidate) and one or more human translations (reference). One such method is the classical edition distance at the word level (WER, for Word Error Rate), based on the Levenshtein distance at word level. BLEU introduced the notion of precision based on n-gram overlap, which was further exploited in other distance-based measures (NIST, ROUGE, and METEOR). These measures express the quality of translations as the similarity with the reference translation(s), although the distance between an excellent human translation and a reference translation might be very high. In our case, the improvement of the translation of connectives might be too small, with respect to the overall n-gram counts, to be detected by such metrics, hence the need to score discourse connectives with a specific metric, while still using e.g. BLEU to control for the overall quality.

Therefore, in order to assess the improvement of discourse connective translation, we define a new evaluation metric named ACT for "Accuracy of Connective Translation".

In a first step, ACT uses a dictionary of possible translations, collected from data and validated by humans. A key point of the metric is the use of a dictionary of equivalents to rate as correct the synonyms of connectives classified by senses.

In a second step, we apply ACT by using alignment information to detect the correct connective translation since a translation can contain more than one connective. If we have wrong alignment information (empty or not equal to a connective), we compare the word position between the source connective or its alignment word (s) in the translation sentence (candidate or reference) and the set of candidate connectives to disambiguate the connectives translation situation.

We evaluate the translation of connectives from English to French/Arabic. The evaluation algorithm is given using the following notations:

- Src: the source sentence
- Ref: the reference translation
- Cand: the candidate translation
- C: Connective in Src
- T(C): list of a priori possible translations of C (from the above dictionaries)
- Cref: reference connective, i.e. translation of C in Ref
- Ccand: candidate connective, i.e. translation of C in Cand.

Table 8 shows the six different possible cases in the first evaluation method. The idea is to compare a candidate translation with a reference translation. We suppose here that there is a connective in the source sentence. We first check if the reference translation contains one of the possible translations of this connective, listed in a dictionary ($T(C) \cap Ref \neq \Phi$). After that, we similarly check if the candidate contains a possible translation of this connective or not ($T(C) \cap Cand \neq \Phi$). Finally, we check if the reference connective found above is equal (case 1), synonym (case 2) or incompatible (case 3) to the candidate connective (Cref=Ccand).

T(C) ∩ Ref ≠ Φ	T(C)∩Cand≠Φ	Cref=Ccand	Decision	
1	1	1	"Same connective in Ref and Cand ==>likely ok !"	1
		~	"Synonym connectives in Ref and Cand ==>likely ok !"	2
		0	"Incompatible connectives"	3
	0	"Not translated in Cand ==> likely not ok"		4
0	1	"Not translated in Ref but translated in Cand ==> indecide, to check by Human"		5
	0	"Not translated in Ref nor in Cand ==> indecide"		6

Table 8: Basic evaluation method without alignment information.

Because discourse relations can be expressed implicitly or not translated, correct translations might also appear in cases 4–6, but they are missed by this metric (which is therefore not lenient).

In total, these different combinations can be

represented by six cases. For each one, ACT prints a specific output message corresponding to the translation situation. These six cases are:

1. *Same connective in the reference and in the candidate translations.*
2. *Synonymous connectives in the reference and in the candidate translations.*
3. *Incompatible connectives in the reference and in the candidate translations.*
4. *The source connective is translated in the reference but not in the candidate translation.*
5. *The source connective is translated in the candidate but not in the reference translation.*
6. *The source connective is neither translated in the reference nor in the candidate translation.*

For case 1 (identical translations) and case 2 (equivalent translations), the ACT metric counts one point, and otherwise zero for cases 3-6. However, one cannot automatically decide for case 5 if the candidate translation is correct, given the absence of a reference translation of the connective. We propose then to check manually these candidate translations by one or more human evaluators. The following example in Figure 1 illustrates case 2, "synonymous connectives".

Csrc = while (whileTEMPORAL)
Cref = bynmA بينما
Ccand = fy Hyn في حين

SOURCE 163: while the group of eight major industrialized countries (g8) and the security council have taken important steps to do this , we need to make sure that these measures are fully enforced and that they reinforce each other .

REFERENCE 163: وبينما اتخذت مجموعة البلدان الصناعية الرئيسية الثمانية ومجلس الأمن خطوات مهمة لتحقيق ذلك، نحتاج إلى التأكد من إنفاذ تلك التدابير بشكل تام وأن يكون يعزز بعضها بعضا.

CANDIDATE 163: وفي حين ان مجموعة البلدان الصناعية الرئيسية الثمانية (مجموعة ال 8) ومجلس الامن قد اتخذت خطوات هامة للقيام بذلك يجب ان نتاكد من ان تكون هذه التدابير تنفيذا كاملا وانها تعزز بعضها بعضا .

Figure 1: Example of ACT case 2.

ACT generates as output a general report, with scores of each case and sentences classified by cases. The total ACT score is the ratio of the total number of points to the number of source connectives, with several possibilities to calculate it. One version is to augment the score by the number of validated translations from case 5.

Three scores are used in the ACT framework, shown in Equations (1)–(3) below. A strict but fully automatic version is ACT**a**, which counts only Cases 1 and 2 as correct and all others as wrong. A more lenient automatic version excludes Case 5 from the counts and is called ACT**a5**. Finally, ACT**m** also considers the correct translations found by manual scoring of Case 5 (noted |Case5corr|).

$$ACTa = (|case1| + |case2|) \Big/ \sum_{i=1}^{6} |casei| \quad (1)$$

$$ACTa5 = (|case1| + |case2|) \Big/ \sum_{i=1}^{4} |casei| + |case6| \quad (2)$$

$$ACTm = (|case1| + |case2| + |case5corr|) \Big/ \sum_{i=1}^{6} |casei| \quad (3)$$

where |case*N*| is the total number of discourse connectives classified in case*N*.

3.2 Meta-evaluation of ACT for French

In order to estimate the accuracy of the first version of ACT (without the disambiguation module based on word alignment and word numeric position information) for English-French, we manually evaluated it on 200 sentences taken from the UN EN/FR corpus, with 204 occurrences of seven discourse connectives (*although, though, even though, while, meanwhile, since, yet*). We counted for each of the six cases the number of occurrences that have been correctly vs. incorrectly scored (each correct translation scores one point). The results were, for case 1: 73/0, case 2: 27/3, case 3: 35/2, case 4: 23/5, and for case 6: 7/0. Among the 29 sentences in case 5, 16 were in fact correct translations.

Therefore, the ACTa score was about 10% lower than reality, while ACTa5 and ACTm were both about 2% lower. This experiment shows that ACT is a good indicator of the accuracy of connective translation, especially in its ACTa5 and ACTm versions.

A strict interpretation of the observed ACT errors would conclude that ACT differences are significant only above 4%, but in fact, as ACT errors tend to be systematic, we believe that even smaller variations are relevant.

Two (opposite) limitations of ACT must be mentioned. On the one hand, while trying to consider acceptable (or "equivalent") translation variants, ACT is still penalized, as is BLEU, by the use of only one reference translation. On the other

hand, the effect on the human reader of correctly vs. wrongly translated connectives is likely more important than for many other words.

In order to estimate the accuracy of ACT by using word alignment, we manually evaluated it on a new subset of 200 sentences taken from the UN EN/FR corpus (different from the first one), with 207 occurrences of the seven discourse connectives. As done for the first version (before adding the disambiguation module) of ACT, we counted for each of the six cases the number of occurrences that have been correctly vs. incorrectly scored. The results were, for case 1: 64/0, case 2: 64/3, case 3: 33/4, case 4: 1/0, and for case 6: 0/0. Among the 38 sentences in case 5, 21 were in fact correct translations. Therefore, the ACTa score was about 10% lower than reality in the initial version of ACT and now is approximately the same, while ACTa5 and ACTm were both about 2% lower and now is 0.5%. Word alignment thus improves the accuracy of the ACT metric.

3.3 Meta-evaluation of ACT for Arabic

We performed a similar evaluation for the English-Arabic version of ACT taking 200 sentences from the UN EN/AR corpus with 205 occurrences of the seven discourse connectives. Results are as follows (correctly vs. incorrectly): for case 1: 43/4, case 2: 73/2, case 3: 27/4, case 4: 19/2, and for case 6: 5/1. Among the 25 sentences in case 5, 9 were in fact correct translations.

Therefore, the ACTa score was about 5% lower than reality, while ACTa5 and ACTm were both about 0.5% lower.

4 Benchmark ACT scores

4.1 Configuration of ACT

ACT can be configured and used with two main versions: with or without the word alignment module. The version with word alignment can be used either without training alignment model using just GIZA++ (Och and Ney, 2000) as alignment tool at the word level, or with training and saving an alignment model. The latter version uses MGIZA++ (a multi-threaded version of GIZA++) trained in a first step on the Europarl corpus (Koehn, 2005) giving an alignment model to be applied on the new data (Source, Reference) and (Source, Candidate). In the following experimentation, we will use the three versions of ACT: ACT without alignment, ACT with alignment but without training the alignment model, and ACT with training the alignment model.

4.2 Data

In all the following experiments, we made use of a set of 2100 sentences taken from the UN EN/AR corpus, with 2206 occurrences of the seven discourse connectives mentioned above (at least 300 occurrences for each one). We developed a baseline SMT system using Moses to translate from English to Arabic.

4.3 Experiments and Results

BLEU is computed here on tokenized, lowercased text for the English data, by using the implementation of the NIST Mteval script v. 11b (available from www.itl.nist.gov/iad/mig/tools/). ACT is computed on tokenized and lowercased text.

Metric	Versions	SMT baseline
BLEU		0.353
NIST		7.517
ACT without disambiguation	ACTa	0.554
	ACTa5	0.643
ACT without training alignment	ACTa	0.563
	ACTa5	0.652
ACT with training alignment	ACTa	0.561
	ACTa5	0.651

Table 9: SMT baseline system, 2100 sentences (without manually checking case 5)

Table 9 contain BLEU, NIST and ACT scores for the SMT system. The 3 configurations of ACT are all used giving each one 3 scores (ACTa, ACTa5). ACTm might be augmented by the number of correct translations from case 5. We didn't check these translations. We just counted the number of occurrences of case 5. This number (303 occurrences) contains correct (approximately 30-50% as shown in section 3.3) and incorrect translations.

5 Conclusion and Future Work

We propose a semi-automatic method to find out Arabic possible translations functionally equivalent to English connectives. It consists of projecting connectives detected on the English side to the Arabic side of a large corpus using alignment information between sentences at the word level. Starting from the result of this method, we build a dictionary of English-Arabic connectives classified by senses.

We developed then a new distance-based metric called ACT, to measure the improvement of a translation model augmented with labels for discourse connectives. In another paper (Meyer et al., 2012), we show that these resulting models (for English-French) perform with BLEU score gains of up to +0.60 points, but the semi-automated evaluation metric ACT shows improvements of up to 8% in the translation of connectives.

This metric applied here on two language pairs (English-French and English-Arabic). Even if it was developed initially for English-French pair, it works well also when applied to English-Arabic. Our goal is also to work towards a multilingual metric.

Acknowledgments

We are grateful for the funding of this work to the Swiss National Science Foundation (SNSF) under the COMTIS Sinergia Project, n CRSI22_127510 (see www.idiap.ch/comtis/)

References

Al-Saif A. Markert K. 2010 The Leeds Arabic Discourse Treebank: Annotating Discourse Connectives for Arabic. In Proc. of LREC, Valletta, Malta.

Banerjee S., and Lavie A. 2005. *METEOR:* An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proc. of the ACL, Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, Ann Arbor, Michigan, US.

Habash N. and Rambow O. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In Proc. of ACL, pages 573–580, Ann Arbor, Michigan.

Iraky K., Zakareya A. F. and Abdelfatah F. 2011. Arabic Discourse Segmentation Based on Rhetorical Methods. International Journal of Electric & Computer Sciences (IJECS-IJENS), vol: 11, n°: 1.

Koehn P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In Proc. of the Tenth Machine Translation Summit, pages 79–86, Phuket, Thailand.

Lavie A. and Denkowski M. 2010. The METEOR Metric for Automatic Evaluation of Machine Translation, Machine Translation, 2010.

Meyer T. and Popescu-Belis A. 2012. Using sense-labeled discourse connectives for statistical machine translation. In Proc. of the EACL Joint Workshop on Exploiting Synergies between IR and MT, and Hybrid Approaches to MT (ESIRMTHyTra), pages 129–138, Avignon, France.

Meyer T., Popescu-Belis A., Hajlaoui N. and Gesmundo A. 2012. Machine Translation of Labeled Discourse Connectives. In the Proc. of AMTA, San Diego, CA.

Meyer T., Popescu-Belis A., Zufferey S., and Cartoni B. 2011. Multilingual annotation and disambiguation of discourse connectives for machine translation. In Proc. of 12th SIGdial Meeting on Discourse and Dialog, pages 194–203, Portland, Oregon, US.

Miltsakaki E., Dinesh N., Prasad R., Joshi A. and Webber B. 2005. Experiments on sense annotations and sense disambiguation of discourse connectives. In Proc. of the 4th Workshop on Treebanks and Linguistic Theories (TLT), Barcelona, Spain.

Nagard R. and Koehn P. 2010. Aiding pronoun translation with co-reference resolution. In Proc. of the Joint 5th Workshop on Statistical Machine Translation and Metrics (MATR), pages 258– 267, Uppsala, Sweden.

Och F., J. and Ney H. 2000. Improved Statistical Alignment Models. Proc. of the 38th ACL, pages 440-447, Hong-Kong, China.

Papineni K., Roukos S., Ward T., and Zhu W. J. 2002. BLEU: a method for automatic evaluation of machine translation. In Proc. ACL, pages. 311–318, Sapporo, Japan.

Pitler E., and Nenkova A. 2009. Using syntax to disambiguate explicit discourse connectives in text. In Proc. of ACL-IJCNLP 2009 (47th Annual Meeting of the ACL and 4th International Joint Conference on NLP of the AFNLP), Short Papers, pages 13–16, Singapore.

Prasad R., Dinesh N. Lee A., Miltsakaki E. Robaldo, L. Joshi, A. and Webber B. 2008. The Penn Discourse Treebank 2.0. In Proc. of LREC, pages 2961– 2968, Marrakech, Morocco.

Idiomatic MWEs and Machine Translation
A Retrieval and Representation Model: the AraMWE Project

Giuliano Lancioni[1]
Roma Tre University
lancioni@uniroma3.it

Marco Boella[1]
University of Rome "La Sapienza"
marco.boella@alice.it

Abstract

A preliminary implementation of AraMWE, a hybrid project that includes a statistical component and a CCG symbolic component to extract and treat MWEs and idioms in Arabic and English parallel texts is presented, together with a general sketch of the system, a thorough description of the statistical component and a proof of concept of the CCG component.

1 Introduction

We present the AraMWE Project[2], a hybrid model able to identify and represent Idiomatic Multi-Word Expressions (IMWE) in Arabic texts. Firstly IMWEs are identified in texts through standard computational quantitative-statistic strategies independent from linguistic knowledge. Then, a formal grammar theory, namely Combinatory Categorial Grammar (CCG), helps to parse and represent the IMWE structure, in order to improve recognition/generation in machine and machine-assisted translation and automatic alignment of specific elements in multilingual texts.

Chapter 2 presents some definitions on IMWE, CCG, Translation Memories and alignment, with related glance on current trends of research. In Chapter 3 the working model and the process flow of AraMWE project will be described, with a special focus on automatic recognition of given IMWE patterns and the strategies we adopted to account for IMWEs in a CCG environment. Chapter 4 gives information on data used and model testing and evaluation, and Chapter 5 closes the paper with some conclusions and an outlook on future developments.

[1] This paper is the result of joint work. However, the authorship can be attributed as follows: 1, 2.1, 2,2, 3.1 and 4 have been written by Boella, 2.3, 3.2 and 5 by Lancioni.
[2] host.uniroma3.it/docenti/lancioni/AraMWE.

2 Subject definitions and related research

2.1 Idiomatic Multiword Expressions

Multi-Word Expressions (MWE) are usually identified in literature with sequences of two or more words that have stronger relationships among themselves rather than with other sentence elements (Cacciari and Tabossi, 1993) or, following another definition, "a multiword unit or a collocation of words that co-occur together statistically more than chance" (Hawwari et al., 2012:24).

Studies on MWEs tend to suggest fluid and smooth classification criteria, which overlap with each other and form a continuum rather than defining sharp subsets (Sag et al., 2002).

The first parameter is semantic in nature and concerns compositionality. On the lower side we find MWEs whose meaning can be guessed by "composing" the meaning of the single elements (e.g. *the president of the republic*). Other MWEs have a medium degree of compositionality i.e. the resulting meaning is not merely a sum of that of the single elements, but somehow still related (e.g. *to carry coals to Newcastle*, which means 'to do something pointless'), up to those MWEs whose meaning has nothing to do with the single elements e.g. the often cited *to kick the bucket* 'to die', or *to spill the beans* 'to reveal secrets' (Cacciari and Tabossi, 1993).

The other main parameter involves morphosyntax: each element occurring in a MWE has a different degree of flexibility, in terms of position (MWE can contain non-MWE elements) and inflection (verb conjugation and noun declension).

Beside criteria of composition and flexibility, MWEs can be further classified according to the main parts of speech involved, e.g. Noun + Noun (NN), Verb + Noun (VN), Verb + Preposition (VP) and so on. These classes seem to have a certain rate of homogeneous behavior involving compositionality and flexibility, e.g. NN seem to be more composi-

tional and less flexible than VN (Cacciari and Tabossi, 1993).

These assumptions clearly don't set clear boundaries and show how difficult it is trying to define which MWE can be fully recognized as idiomatic. Idiomaticity seems obviously connected with low compositionality and relatively low flexibility (in positioning especially), but a clear definition is far to be outlined (Pawley, 1983), even if a long tradition of studies assigns to "idiomaticity" just the same meaning of "compositionality" (see for example Diab and Bhutada, 2009). For the purposes of our work, we provisionally call Idiomatic MWEs those multi-word expressions semantically non-compositional and syntactically non-conforming (see also Kavka and Zybert, 2004).

Related work: Concerning NLP approach to MWEs in Arabic, recent studies focus on two main directions, the construction of annotated repositories of MWEs and the automatic detection and extraction of MWEs from texts. Approaches for the first issue vary from those fully unsupervised (Cook et al., 2007) to more recent hybrid models that include supervised procedures to improve size and correctness of the list (Hawwari et al., 2012; Diab and Bhutada, 2009). Several works concern instead the automatic extraction of MWEs, with strong statistical approaches (Al Khatib and Badarneh, 2010; Moirón et al. 2006). Other recent models focus on parallel strategies to feed models with linguistic or statistical information needed to discern MWEs, especially for nominal ones (N+N) (Attia et al., 2010).

2.2 Translation memories and alignment

In the field of machine-assisted translation the collections of bilingual texts known as Translation Memories (TMs) aid human translator by providing sentences or larger text chunks in a given language, together with the 'aligned translation in another language, or other languages. Through strict or fuzzy search a translator can look up in the TM for the best match for the word context needed to perform correct translation. The employ of TMs is mainly as commercial and professional tool, and TM implication in computational and corpus linguistics was scarcely investigated, nevertheless some recent studies aim to reduce the size of aligned text chunks by using parsing systems, from sentences to sub-sentence elements, with the goal to get a complete aligned, cross-referenced bilingual parallel corpus (Lagoudaki, 2006).

Related Work: Many studies propose models to deepen TM alignment, in order to pair not only paragraphs and sentences, but also phrases, words and even word constituents (Simard, 2003). Among works that treat TMs specific to less represented languages focusing on an unsupervised approach, Chuang et al. (2005) show how to build a Chinese-English TM integrating statistical and linguistic information, and trying to analyze and align sub-sentence chunks. Concerning TMs covering Arabic, beside some commercial multilingual products in which Arabic is just one of the several languages provided, the most interesting example of an Arabic-English TM is Meedan (2009), an open access collection of several thousand paired text chunks extracted from Arabic newswires. Its structure is the simplest, providing just Arabic sentences paired with English translations, without any alignment of sub-sentences.

2.3 Combinatory-Categorial Grammar (CCG)

The choice of CCG as a grammatical paradigm to analyze and automatically translate idioms is based upon several grounds: (i) it is a perfectly formalized grammatical paradigm; (ii) some very performing implementations, such as OpenCCG (White, 2012; Bozşahin et al.; 2012), are available, with both parsing and generation capabilities; (iii) the lack of a theoretical status for phrase structure allows for highly unortodox structures to be represented, e.g. coordination among elliptical constructions (Steedman, 2000; Steedman and Baldridge, 2005), which fits well the complex nature of idioms requirements as far as phrase structure is involved; (iv) the combination of a very basic categorial apparatus with infinitely many complex categories and attributes allows for a smooth transition between open constructions, partially frozen collocations and more or less rigid idioms.

In the original, simplest version, A[jdukiewicz] B[ar-Hillel] Calculus (Bar-Hillel, 1953), a single "rule", functional application, is included: a complex category matches another element to its left or its right (according to the direction of the final slash) to form a larger category where the matched element is "erased" from the list of missing arguments. The function in the semantics of the complex category is applied to the semantics of the matched argument.

the po-liceman	departed
NP: **P**	S\NP: λx.**go**(x)
S: **go(P)**	

Example 1

the po-liceman	saw	the boy
NP: **P**	S\NP/NP: λxλy.**see**(y,x)	NP: **B**
	S\NP: λy.**see**(y,**B**)	
S: **see**(**P**,**B**)		

Example 2

AB Calculus is weakly equivalent to CF grammars (same generative power, possibly different analyses). This limitation does not allow the analysis of known phenomena that are slightly beyond strict context-freeness (e.g., cross formations in Dutch and Swiss German) and makes it difficult to handle unbounded dependencies. Since Curry & Feis (1958) "curried" operators (functional composition, type raising, crossed composition) have been introduced into the machinery of CG, which results in Combinatory Categorial Grammar (Steedman, 2000).

Related Work: Thanks to its very clear formal properties, CCG has been used for some very large implementations in parsing and generation. In particular, the CCGbank project (Hockenmaier and Steedman, 2005) translated the whole of Penn Treebank into a corpus of CCG derivations; the C&C CCG parser and supertagger, together with the Boxer computational semantics tool (Curran et al., 2007), have been explicitly designed for large-scale NLP tasks; OpenCCG, the OpenNLP CCG Library (Baldridge et al. 2007), implements a parser and a realizer with supertagging and hypertagging modules in the framework of multi-modal extensions to CCG (Baldridge and Kruijff, 2003). Several large grammars have been implemented in OpenCCG, including Moloko, a grammar oriented towards parsing and realization in human-robot interaction (Kruijff and Benjamin, 2012). However, with all their theoretical and empirical advantages CCG models have virtually never have been applied to the analysis of idioms nor to MT applications. The reason for this probably lies in a certain hesitation by linguists in the CCG framework to tackle language universals and in the idea that CCG semantic representation is best strictly coupled to its syntactic counterpart, which seems to make the treatment of wildly different structures that convey the same "meaning" in natural languages rather unlikely. As the proof-of-concept application presented in 3.2. shows, this is not necessarily the case.

3 The model and its implementation

The model we propose, given a list of IMWEs enriched by some semantic information, searches for them in collections of non sub-sententially aligned bilingual text (namely TMs), trying to pair each Arabic IMWE with the related translated chunk via the CCG representation module, that builds a syntactic-semantic representation of the matching IMWEs. The modular structure of the model will allow future developments, especially for the CCG component, which can be ideally extended in order to parse the entire TM and to get fully aligned bilingual versions.

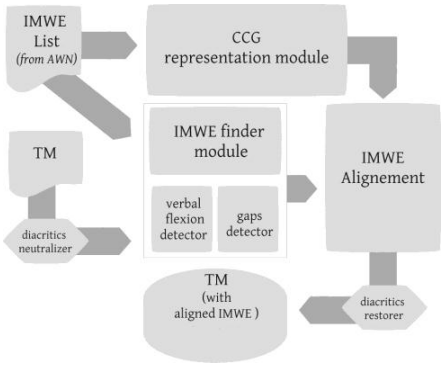


Figure 1: Model's process flow

3.1 Setting the MWE list and the pattern matching strategy

Since the aim of our model is not automatic extraction of MWEs, but rather testing alignment through a CCG interpretation, the IMWE list is a pre-existent input, but the condition is that every lexical entry must be previously associated with some semantic information (synonyms, English translations, ontological classification), usually available in networks such as Arabic WordNet (AWN: Black et al., 2006)). The main advantage in benefiting of data extracted from a lexical network is to have not only standard translations, but also all available MWEs in the target language. The example below shows a typical MWE entry used as an input:

intaqala 'ilā al-rafiq al-'a'lā (['die', 'decease', 'perish', 'go', 'exit', 'pass away', 'expire', 'pass', 'kick the bucket', 'cash in one's chips', 'buy the farm', 'conk', 'give-up the ghost', 'drop dead', 'pop off', 'choke', 'croak', 'snuff it'], ('die_v_1', 'Death'))

The length of the list is not very important, as the main task for this work is to have semantic data included in order to test the CCG module. Obviously

the model could benefit of other MWE sources, such as dictionaries, exhaustiive MWE repositories (see Hawwari et al., 2012), other network ontologies (e.g. Arabic VerbNet, FrameNet) or *ad hoc* lists built on web multilingual cross-referenced resources, such as Wikipedia.

At this stage of the model implementation we chose to focus on MWEs that contain at least a verb, in order to experiment more complex argument representations in CCG module. Almost all these MWEs share a low degree of compositionality and a certain morphosyntactic flexibility.

The other main input is obviously the TM, in which we would align the MWEs that match patterns in the list.

As it is known, the Arabic writing system includes a diacritization system for marking short vowels and consonant lengthening, but this system is rarely used in contemporary texts. However, since a partial diacritization is always possible even in contemporary writings, it can generate lots of false negatives if it is not taken into account. A small, independent module is therefore foreseen to neutralize full or partial vocalization in both MWEs list and TM processing and at the output stage to restore the original configuration.

Both inputs are then processed in a module that select the entries contained in MWE list as patterns to be matched in the TM. Since MWEs in the TM can have various degree of flexibility (namely verbal conjugation and a certain degree syntactic mobility of the constituents), two sub-modules has been conceived.

The first one accounts for morphological flexibility, but works in the lightest possible way, avoiding the need of new linguistic information. This is achieved by selecting in the verbal MWE pattern (always conjugated at past tense, third person masculine singular) those letters that are preserved in every conjugated form, i.e. the consonants (both belonging to the root and marking stems, e.g., *istaslama* > **s*t*s*l*m**, which is common to every conjugated form, such as *yastaslimu*, *istaslamna*, and so on). To deal with irregular verbs, the semi-consonants *w* and *y*, together with the *'alif* symbol are also ignored. In the next chapter it will be shown that this sort of brute-force method seems to provide better results than the employment of an external lemmatizer, namely the Buckwalter morphological analyzer (Buckwalter, 2002). Such tool appears to be instead more effective as a further strategy in refining results of the brute-force method, but this hypothesis was not yet tested with standard evaluation criteria.

The second sub-module simply allows to find MWE constituents in the target text even if they are intercalated with non MWE elements, by using gap detecting algorithms modeled on regular expression syntax.

Finally, the matching MWEs retrieved in the Arabic section of the TM are automatically tagged with the related source information contained in the original MWE list, in order to be processed by the CCG module.

3.2 Representation through CCG

As a proof of concept for the approach in representing syntax and semantics of idioms in the framework of a bilingual, bi-directional Arabic-English machine translation, two proof-of-concept (POC) grammars, one for each of the languages, were written in OpenCCG. Both grammars translate between surface forms and semantic representations and the other way round, being able to parse and generate from the same architecture. No direct language-to-language mechanism is included, and machine translation is rather a by-product of single-language parsing and generation facilities that share a common semantic representation.

The semantic representation avails itself of the dual representation level in OpenCCG: each non-purely functional word is grounded to a predicate and a class. The predicate is the main semantic value of a word and works as a key to parsing and, especially, generation. The class serves to match semantic restrictions on arguments: e.g., actors are animate.

In order to have a reasonably universal, or at least not excessively language-biased, semantic component, predicates are chosen among WordNet synsets and classes among SUMO concepts (Niles and Pease, 2001). These choices were induced by several reasons: on the one hand, WordNet (Miller, 1995) is perhaps the single most widespread lexical resource publicly available and a de facto standard in language technologies, alignments to it are available for many other resources —such as VerbNet, FrameNet, Wiktionary, SUMO and, most importantly, Arabic WordNet among many localized versions of the lexical database,— and it is a very practical choice for a universal semantic component; the unavoidable linguistic bias towards English will be overcome in further developments by treating WordNet as the main source for an International Language Index

(ILI: Vossen, 1998) together with other sources: this is what already happens in many localized WordNets, e.g. cultural-oriented concepts added in Arabic Wordnet are already assigned an ID distinct from the English WordNet.

On the other hand, the choice of SUMO concepts as a source for classes, though perhaps less straightforward, is reasonable as well; even if the roughly 3800 SUMO ontologies to which the 117k WordNet synsets are mapped are way too many for most reasonable linguistic tasks (VerbNet 3.2 uses only 36 selectional restrictions for 6031 verbs), the use of a larger ontology can be useful for more specialized lexical selection (e.g., the verbs in VerbNet class 38, animal sounds, all have the restriction [+animal] on the agent, but it is more reasonable also in linguistic terms to have a stricter restriction: for instance, only cats tend to meow) and —perhaps more importantly— the representation of the semantic component through ontologies with a rich axiomatization such as SUMO can be the input to further components, for instance a reasoner.

The POC grammar has a limited number of synsets, 5 nominal and 7 verbal ones, expressed by 18 English and 18 Arabic lexemes (including MWEs). The (rather large) subset of SUMO that encodes the corresponding classes, together with relevant WordNet synsets and English and Arabic lexemes, are shown in Figure 2 (arrows mark subclass relations, instanced classes have a light blue background, general classes for nouns and verbs are in salmon red and WordNet synsets are within boxes, with English and Arabic lexemes in italics).

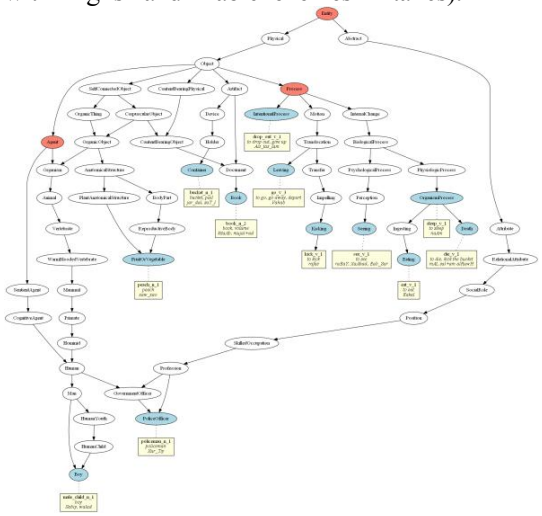


Figure 2: The network of SUMO classes, WordNet synsets and lexemes of the POC grammar

Despite its limitations, this POC addresses a number of potentially thorny issues in bilingual MT. First, the strongly lexical nature of CCG allows syntactic differences between English and Arabic to be abstracted away from semantic representation. E.g., the only relevant difference between Arabic and English intransitive verbs is the direction of the slash (basically, S/N in Arabic and S\N in English; we disregard here topic-initial sentences in Arabic, that are probably best analyzed as XVS structures according to the standard analysis in the Arabic grammatical tradition).

The key to extend the CCG approach to increasingly noncompositional lies in the more or less standard treatment of case-marking prepositions: if a verb requires a complement introduced by *to*, the latter does not contribute to the composition of the semantic representation; rather, it merely "checks" a syntactic feature that is needed for the derivation to continue.

In the same vein, the main significant element in an idiom is lexically assigned the semantic representation, while less significant elements are given a syntactic, checking function which is nevertheless necessary in order to let the derivation go on.

As an example, let us see how the system derives two idioms, one English and one Arabic, that Arabic WordNet considers equivalent to 'to die' in the meaning 'pass from physical life and lose all bodily attributes and functions necessary to sustain life', *kick the bucket* and *sal+am alruwH*[3], respectively (see entry example in 3.1). The English idiom admits of two reading, the idiomatic one and the less likely, but admissible, literal reading 'to give a kick to the pail'.

The POC grammar attributes the key role in the idiom to the verb *to kick* and uses the NP *the bucket* as a checking element. While debatable, this choice is not entirely arbitrary: on the one hand, it is *ceteris paribus* preferable to attribute the verb the key semantic role, since it already has the key role in the syntactic derivation; on the other hand, the shortened form *to kick* is attested in the meaning of the idiom, even if it is not recorded in WordNet (it is recorded in the English Wiktionary and in meaning I.b of $kick_{v1}$ in the OED).

The idiomatic and the literal derivations are shown in Figure 3 and Figure 4 respectively:

[3] The Arabic transcription is a 7-bit ASCII compliant version of the Buckwalter. We adopt a simplified morphology, without the final declension vowels that are usually omitted in everyday Modern Standard Arabic

Figure 3: derivation (idiomatic reading)

Figure 4: derivation (literal reading)

Two very distinct semantic representations are get by very similar syntactic derivations. The details of the semantic representations are in Figure 5 and Figure 6 respectively.

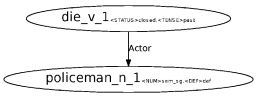


Figure 5: semantic representation (idiomatic reading)

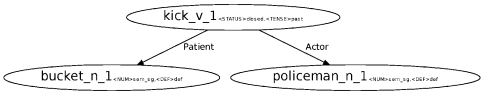


Figure 6: semantic representation (literal reading)

The Arabic version *sal+am alruwH* has basically the idiomatic reading only. The literal reading of 'to deliver the soul (to God)' is improbable enough, and close enough to the idiomatic reading, to have been excluded in our POC grammar (however, it might be included without altering the nature of the results). Here is the derivation for *sal+am AlXur_Tiy AlruwH* 'the policeman delivered his soul', i.e. 'died':

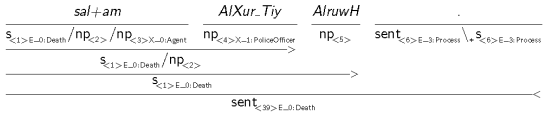


Figure 1: derivation of *sal+am AlXur_Tiy AlruwH*

The most striking feature of this derivation is that notwithstanding its radical syntactic dissimilarity from its English counterpart, it produces exactly the same semantic representation in Figure 6 above. This is the meaning of MT in this model: two or more sentences translate one another when they have the same semantic representation.

If we feed the English realizer with the representation in Figure 6 above we get the following sentences (in no particular order, unless we add scorer to the realizer, see White, 2012 for details):

the policeman died .
the policeman kicked the bucket .

If we feed the same representation to the Arabic realizer, we get

maAt AlXur_Tiy .
sal+am AlXur_Tiy AlruwH .

In both case, the first sentence is a literal, the second one an idiomatic, equivalent of *the policeman died* in the two languages.

On the other hand, if we feed the realizers with the representation in Figure 5, we get:

the policeman kicked the bucket .
the policeman kicked the pail .

for English, and:

rafas AlXur_Tiy Aljar_dal .
rafas AlXur_Tiy AlsaT_l .

for Arabic. In both cases, we have equivalents for the literal meaning of 'the policeman gave a kick to a (real) bucket', with different lexemes for 'bucket'.

This POC, notwithstanding its limitations, shows a series of interesting features: (1)identical meanings are captured despite of very different syntactic derivations, and different meanings are captured for the same input strings; (2) a language-independent representation of meaning is obtained, which can feed other components (reasoners etc.); (3) MT is a by-product of the parsing and realizing: translating in this model is not structurally different from paraphrasing (which is one of the main uses in current implementations of OpenCCG); (4) the system can be extended to other languages without the need to implement language-to-language grammar couples (the coupling is obtained through identity of semantic representations).

4 Testing model and results

4.1 Data and instruments

The source for the employed IMWE list is AWN (see also Rodrigo et al. 2008). Relatively small in size (it contains around 11,000 synsets), AWN utilizes the Suggested Upper Merged Ontology (SUMO) as a common interface to dialogue with previously developed wordnets.

We used two different TMs to test the model, one in Contemporary Arabic, the other in Classical Arabic. The first one is the Arabic-English Meedan

Translation Memory v.10 (Meedan 2009), which contains 59861 paired textual excerpts, mostly sentences, for around one million words. The source is declared to be newswires in Arabic.

The second one consists of our provisional version of a parallel Arabic-English corpus based on al-Bukhārī's collection of Hadiths. This corpus is still under development (and results are not still published) and at the present stage it only pairs the full *matn* (content) part with the correspondent English translation, without sentence segmentation. The number of paired *matn*s is 7305, with 382,700 words.

4.2 Testing and Results

At the beginning, all verbal MWEs have been extracted from the AWN verbal synsets, by searching for all entries containing at least one blank space surrounded by words. From the 666 resulting MWEs we omitted those with the pattern Verb + Preposition, as generally more compositional and less idiomatic. The resulting list was populated by 387 entries. To each entry its form without diacritics was then automatically associated. Both target TMs has been treated in the same way, by neutralizing any diacritization.

The MWE list fed the set of patterns to be searched in TMs (Meedan and Hadith corpus), with an interaction with related sub-modules to neutralize verbal conjugation and syntactic flexibility. The results of the MWE identification process are briefly shown in Table 2.

The automatic alignment of Arabic MWEs with correspondent English chunks was performed by using the drafted CCG module (results in Table 3.).

Results of both MWE retrieval in TMs and alignment through CCG have been submitted to standard evaluation practice. The two TMs were divided in training and testing sections, through division of each corpus in a training (85%) and testing (15%) part; the latter is currently still relatively small in consideration of the homogeneity of the corpus and the need to manually annotate the test sentences.

	MWE Retrieval		**CCG Alignement**	
	Meedan TM	Hadith TM	Meedan TM	Hadith TM
Error rate	20.57	15.35	8.07	10.9
Precision	85.24	88.29	95.68	93.23
Recall	94.19	96.36	96.25	95.87
F_1	89.49	92.15	95.96	94.53

Table 2 – Summary of results

Concerning MWE retrieval, a manual screening of the testing sample showed a consistent error rate (20.57% for Meedan TM and 15.35% for Hadith TM). However, considering the high number of false negatives (14.76% for Meedan TM and 11,71% for Hadith TM) compared to the small rate of false positives (5,81% for Meedan TM and 3,64% for Hadith TM) , the error rate seems to be mostly due to the relatively small size of the MWE list used as input (which can be easily extended) rather than to the effectiveness of the retrieval module and related sub-modules.

The results of the CCG processing and alignment of retrieved MWEs show instead that the model is highly efficient in pairing Arabic MWEs with related English translations.

5 Conclusions

The AraMWE project aims to bring together statistical analysis and extraction of MWEs in Arabic-English bilingual texts with MT and the building of a semantic representation of sentences containing idioms in the two languages. Although the project is still in its initial stage, preliminary results show the possibility to perform the retrieval stage of the task automatically and in order to feed a symbolic component whose general features have been successfully designed and tested.

Next stages in the project will involve the implementation of an Arabic-English bilingual grammar beyond the POC state in order to cope with a reasonable high percentage of sentences containing MWEs in aligned texts. The final aim of AraMWE is to build a hybrid system where a symbolic CCG-based core grammar is able to analyze, and to provide a semantic representation for, as large as possible an amount of relevant cases, by developing in parallel a statistical component which acts as a back-off mechanism for cases unrecognized by the symbolic component.

6 References

Al Khatib, Khalid, Amer Badarneh. 2010. Automatic Extraction of Arabic Multi-Word Terms. In Proceedings of IMCSIT-2010, 411-418.

Attia, Mohammed, Antonio Toral, Lamia Tounsi, Pavel Pecina and Josef van Genabith. 2010. Automatic extraction of Arabic multiword expressions. In Proceedings of LREC-2010, Valletta, Malta.

Baldridge, Jason and Geert-Jan M. Kruijff. 2003. Multi-Modal Combinatory Categorial Grammar. EACL-03, 211-218.

Baldridge, Jason, Sudipta Chatterjee, Alexis Palmer, and Ben Wing. 2007. DotCCG and VisCCG: Wiki and Programming Paradigms for Improved Grammar Engineering with OpenCCG. In Proceedings of the Workshop on Grammar Engineering Across Frameworks. Stanford, CA.

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetical notation for syntactic description. Language 29: 47–58.

Black, William, Sabri Elkateb, Horacio Rodriguez, Musa Alkhalifa, Piek Vossen, Adam Pease and Christiane Fellbaum. 2006. Introducing the Arabic WordNet Project, in Proceedings of the Third International WordNet Conference, Sojka, Choi, Fellbaum and Vossen eds.

Boulaknadel, Siham, Beatrice Daille, Driss Aboutajdine. 2009. A multi-word term extraction program for Arabic language. LREC 2008, 630–634

Bozşahin, Cem, Geert-Jan M. Kruij and Michael White. 2012. Specifying Grammars for OpenCCG: A Rough Guide.

Buckwalter, Tim. 2002. Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, Philadelphia.

Cacciari, Cristina and Patrizia Tabossi, eds. 1993. Idioms: processing, structure, and interpretation. Lawrence Erlbaum Associates, Hillsdale, NJ.

Chuang, Thomas C., Jia-yan Jian , Yu-chia Chang , Jason S. Chang. 2005. Collocational Translation Memory Extraction Based on Statistical and Linguistic Information. *Computational Linguistics and Chinese Language Processing*, 10 (1), 329-346.

Cook, Paul, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling their weight: Exploiting syntactic forms for the automatic identification of idiomatic expressions in context. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, Prague, Czech Republic. Association for Computational Linguistics, 41–48.

Curran, James R., Stephen Clark, and Johan Bos (2007). Linguistically Motivated Large-Scale NLP with C&C and Boxer. Proceedings of the ACL 2007 Demonstrations Session (ACL-07 demo), 33-36.

Curry, Haskell B. and Robert Feys. 1958. Combinatory Logic: Vol I. Amsterdam: North Holland.

Diab, Mona and Pravin Bhutada. 2009. Verb noun construction MWE token supervised classification. In *Workshop on Multiword Expressions (ACL-IJCNLP)*, 17–22.

Hawwari, Abdelati, Kfir Bar, Mona Diab. 2012. Building an Arabic Multiword Expressions Repository. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Jeju, Republic of Korea, 12 July 2012. Association for Computational Linguistics, 24–29.

Hockenmaier, Julia and Mark Steedman. 2005. CCGbank: User's Manual. Technical Report MS-CIS-05-09, Department of Computer and Information Science, University of Pennsylvania, Philadelphia.

Kavka, Stanislav and Jerzy Zybert. 2004. Glimpses on the History of Idiomaticity Issues. In *SKAZE Journal of Theoretical Linguistics*, 1, 54-66.

Kruijff Geert-Jan M. and Trevor Benjamin. 2012. Documentation for the MOLOKO CCG grammar (v6). The DFKI Language Technology Lab.

Lagoudaki, Elina. 2006. Translation Memory systems: Enlightening users' perspective. Key finding of the TM Survey 2006 carried out during July and August 2006. Translation Memories Survey 2006. London, Imperial College.

Miller, George A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* Vol. 38, (11), 39-41.

Meedan. 2009. Meedan v.10. Arabic-English Translation Memory. Meedan, San Francisco, CA.

Moirón, Begoña Villada and Jörg Tiedemann. 2006. Identifying idiomatic expressions using automatic word-alignment. In the Workshop on Multiword Expressions in a Multilingual Context, EACL-06, Trento, Italy.

Niles, Ian and Adam Pease. 2001. Towards a Standard Upper Ontology. In Proceedings of FOIS-2001. Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.

Pawley, Andrew. 1983. "Two puzzles for linguistic theory: nativelike selection and nativelike fluency." In: J. C. Richards and R.W. Schmidt (eds.), Language and Communication. London: Longman, 191-225.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. CICLing-2002: 1-15.

Simard, Michel. Translation Spotting for Translation Memories. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts: data driven machine translation and beyond* – Vol. 3, pp. 65-72.

Steedman, Mark and Jason Baldridge. 2005 Combinatory Categorial Grammar. In R. Borsley and K. Borjars, eds, Non-Transformational Syntax.

Steedman, Mark. 2000. The Syntactic Process, MIT Press. Boston, MA.

Vossen, Piek, ed. 1998. EuroWordNet: a multilingual database with lexical semantic networks for European Languages. Kluwer, Dordrecht.

White, Michael. 2012. OpenCCG Realizer Manual.

Developing an Open-domain English-Farsi Translation System Using AFEC: Amirkabir Bilingual Farsi-English Corpus

FattanehJabbari, SomayehBakhshaei, Seyed Mohammad Mohammadzadeh Ziabary, ShahramKhadivi
Human Language Technology Lab
Department of Computer Engineering and Information Technology
Amirkabir University of Technology
Tehran, Iran
fjabbari@ce.sharif.edu, {bakhshaei, mehran.m, khadivi}@aut.ac.ir

Abstract

The translation quality of Statistical Machine Translation (SMT) depends on the amount of input data especially for morphologically rich languages. Farsi (Persian) language is such a language which has few NLP resources. It also suffers from the non-standard written characters which causes a large variety in the written form of each character. Moreover, the structural difference between Farsi and English results in long range reorderings which cannot be modeled by common SMT reordering models. Here, we try to improve the existing English-Farsi SMT system focusing on these challenges first by expanding our bilingual limited-domain corpus to an open-domain one. Then, to alleviate the character variations, a new text normalization algorithm is offered. Finally, some hand-crafted rules are applied to reduce the structural differences. Using the new corpus, the experimental results showed 8.82% BLEU improvement by applying new normalization method and 9.1% BLEU when rules are used.

1 Introduction

Statistical Machine Translation (SMT), the most promising MT approaches, is producing acceptable translation for some languages, but not for all language pairs because of some challenges. For example, since it requires a big amount of training data, the translation quality is low for those languages with scarce resources. The problem is also more critical for morphologically rich languages. Farsi is an instance of such languages which has insufficient size of existing parallel corpora, in addition to its rich morphology. Although its morphology is not as rich as Arabic but is richer than most of the languages like English [1]. So, preparing a SMT system for the English-Farsi language pair, results in weak translation quality using small training data, as the previous researches on English-Farsi SMT systems. Considering the related problems in English-Farsi translation, we try to develop a more qualified system. To this end, we first generated a large open-domain parallel corpus, Amirkabir Bilingual Farsi-English Corpus (AFEC). The produced corpus can be considered as the best bilingual parallel English-Farsi corpus according to its size, quality, and domain generality in the news issues.

Furthermore, another difficulty rises when translating from/to Farsi texts, which is the existence of different written forms for each character in Farsi. To remove this character abusing, we offered a new algorithm for text pre and post processing called Essential for Statistical Machine Translation (E4SMT) which uses a high speed character-based algorithm for simultaneous normalization, tokenization and detection of special tokens (e.g. Numbers, Dates,

Abbreviations, etc) by reviewing whole text in a single pass.

Finally, we try to handle another complication of English-Farsi language pair which is the effect of differences in the grammatical structures of English-Farsi language pair. For example, the part of speech order in a Farsi sentence is: Subject Object Verb (SOV), but it is SVO in English. This variation causes to long displacements which are hard to detect by many of the reordering models (since most of them consider the local short distortions). To moderate the differences in words order, we applied some hand-crafted rules which change the order of words in the source language to match the structure of the target side. For this task, we have extracted some manual rules making use of part of speech tags.

The previous considerable researches on English-Farsi languages are [2], [3], and [4] which are the first attempts for making a SMT system for English-Farsi language pair. These researches are developing Automatic Speech Recognition (ASR) systems and try to run speech to speech translation systems, with an essential SMT component as an inner core. They have used either a small corpus, or a limited-domain one. For instance, [2] is a speech to speech translation system in the medical care domain. Thus, our system outperforms the previous SMT systems for English-Farsi language pair since it uses a larger open-domain corpus. Recently, some new experiments are reported like [5] which offer how to build SMT system from limited resources. They have used normalization just on the English side according to the NIST standard table of normalization rules. Compared to this work we have offered a novel dynamic normalization algorithm for both English and Farsi sides. [6] uses a 130K lined corpus with 2.8M running words. This paper has improved the reordering model with a novel idea for Farsi-to-English SMT system. [7] offered a direct search for minimizing error rate for parameter optimization in Farsi-to-English SMT system, instead of MERT algorithm [8], using the corpus size of about 739K line. The corpus we collected in this research is more noticeable than the existing corpora in its size, domain generality, and the numbers of words it covers.

The remainder of this paper is as follows. We describe our corpus generation method in the second section. Then the data normalization scheme is explained in the third section. The forth part is about manual rules. Experiments are explained in section 5. Finally, section 6 concludes the paper.

2 Corpus Gathering

There are two main approaches to create a new corpus: 1) using automatic tools for document aligning, 2) by means of human translators. In this research, both of these methods are used. First, we crawled the web and extracted as much data as possible including parallel, comparable and monolingual texts. In addition to web pages, we used other resources like translated books, software manuals, subtitle-films, multilingual constitution of some countries, etc. Among the gathered data, a small volume was completely parallel, while the rest were the comparable documents.

Bilingual Corpus		Line Number	Singleton	Running Words	Lexicon
Central Asia	English	84807	27722	1971667	61565
	Farsi	84807	18735	2152752	41191
Ted	English	66534	10921	628963	24590
	Farsi	66534	14724	668450	29382
News	English	282227	61537	6993837	135365
	Farsi	282227	75225	7494634	135284
Verb-mobil	English	23145	1039	249356	2763
	Farsi	23145	2414	216577	5283
Misc	English	141602	54319	3343737	105713
	Farsi	141602	44634	3541859	82579

Table 1. Statistics of generated corpora

The qualified comparable data was selected and document aligned with aligner tools. We have used HunAlign [9] and Microsoft aligner [10]. Since these tools are not customized for Farsi language, many parts of the automatically aligned corpora were in such a bad condition that we ignored them. Thus, the produced data was not as much as we needed. We continued the work by translating some part of the documents by the help of human translators. The statistics of each created corpus are shown in Table 1.

In the following section, we will describe much about each of these prepared corpora and the existing ones.

2.1 The automatically aligned corpora

- **CentralAsia** - The first corpus named Central Asia is extracted from Central Asia news website: http://centralasiaonline.com. This website reports news in different languages such as Farsi, English, Urdu, Pashtu, but we have used only Farsi-English parts. It has 84K lines, with about 1.9M words in the English side and 2M words in the Farsi side. Its domain is news domain.

- **Ted** - Ted corpus is the subtitles of the ted website movies: http://www.ted.com/talks. Since the different subjects are presented in this website, the corpus is open-domain. The size of corpus is about 66K lines, with 620K words in the English side and 660K words in the Farsi side.

2.2 Human translated corpora

- **News** - This corpus is the monolingual documents downloaded from news websites such as CNN, BBC, etc. Its volume is about 280K lines with about 6.9M words in the English side and 7.4M words in the Farsi side.

- **Misc** - Misc corpus is a bunch of miscellaneous documents translated by human translators. It has general domain with size of 140K lines and 3.3M words in the English side and 3.5M words in the Farsi side.

- **Verbmobil** - Is a part of English side of Verbmobil project corpus [11] which includes some tourists' conversations about time scheduling and appointment settings and is translated by human translators. This dataset includes 23K lines in both sides, 249K and 216K words in Farsi and English sides respectively.

2.3 The existing corpora

We used some existing corpora in addition to the corpora that we made, which are:

- **Pen** - An existing corpus with about 30K lines. Its domain is news [12].

- **Elra** - An existing corpus with 50K lines which has the news domain [13].

- Another Farsi-English existing corpus is Tehran University Corpus [14]. This corpus is extracted from subtitle films. Its domain is general and sentences are transcriptions of spontaneous speech. The size of this corpus is 612K. The corpus is noisy, so we did not use it in or works.

- 20K transliterated names for further improvement was produced and added it to our integrated corpus [15].

2.4 The AFEC corpus

By integrating all generated and existing corpora, we produced our large corpus. The information of this new corpus is mentioned in Table 2. The lines number of this corpus is about 700M. This corpus covers 14.7G words of English sides and about 15.8G of Persian side.

Bilingual Corpus		Line Number	Singleton	Running Words	Lexicon
AFEC	English	700916	139041	14764413	267717
AFEC	Farsi	700916	133413	15807981	238571

Table 2. Statistics of AFEC corpus

3 DATA NORMALIZATION

Farsi has an important challenge in its written form. This dilemma originates from existence of different ASCII codes for each Farsi written character since there is not a standard format for Farsi written text. Moreover, some characters are misplaced by their Arabic format, because of their similar appearance, for example using "ئ" or "ي" instead of "ی" . We propose a text pre and post processing tool incorporated with an interactive text normalizer to remove this complication we called this tool E4SMT (Essential for Statistical Machine Translation).

The proposed tool is incorporated with a bunch of plugins where each one monitors the occurrence of a specific token. These specific tokens are something like numbers, dates, abbreviations, etc

which must be treated different from other parts of the context or maybe does not need to be translated. Also, a built-in character normalizer module normalizes different character representations to be uniform. The innovative characteristic of the algorithm is the ability of processing, normalization and tagging the whole text in a single pass. By visiting each character, along with normalizing it, all of the plugin modules will process it and cache in case it is a valid character in the sequence. Whenever a plugin module detects new valid token, it will report it to be tagged. Plugins are controlled by a plugin manager and could be deactivated and/or prioritized to change tool behavior in case of similar tokens detection by different plugins.

E4SMT has been developed using C++ in a cross platform scheme thanks to Nokia Qt framework [16] and can be used as a standalone application, as a web service, and also can be integrated to other tools using its API. This tool has many features which are not used in the pre and post-processing parts but used in corpora generation and maintenance. Built-in modules and plugins are incorporated with external configuration files and tables which eases the use, maintenance and enhancement of the tool. Currently, the following built-in features and plugins are developed and activated:

- Character normalizer: This is a built-in feature which works in two interactive and non-interactive modes to convert each Unicode character to a uniform representation
- Built-in tokenizer and tagger: These will tokenize input text and tag specific tokens using plugins. Inline XML (IXML) is used for tagging. IXML tags will be removed in post-processing pass.
- URL plugin: This recognizes URL addresses in the text and tag them
- Email plugin: Similar to the URL plugin, this one recognizes e-mail patterns.
- Suffix plugin: Check for suffixes such as apostrophes by using the suffix tables and some manual rules to exclude them from tokenization process.
- Number plugin: This part recognizes and tags different number types in the text including general numbers, currencies, weights, etc.
- Abbreviation plugin: Recognizes and tags abbreviation words in the text using a dictionary and also some predefined rules. Abbreviations will be converted to their equivalent in post-processing of translated text.
- Transliteration plugin: This plugin will transliterate Name Entities recognized (NER) in input text.
- Virastyar Plugin: This one is a special plugin used for post-processing and correction of punctuations and dictation problems in the translated text.

One of the most important features of the E4SMT tool which caused high improvement in translation results is the normalization feature. At first, we had used a static mapping table to normalize characters both in Persian and English texts. But we found that there are many other unrecognized or multiform characters in texts (especially Farsi texts) downloaded from news agencies which need to be normalized. So, we developed an interactive normalizer which will ask for user decision on any new seen character. Valid decisions are:

- Keep it: the input character must be moved to output without any change
- Remove it: null will be passed as output
- Change it: another character will be replaced.

User decisions will be stored in normalization table and used next time the character is seen both in interactive and non-interactive use of the tool. Now, our normalization table has more than 600 entries covering whole AFEC corpora.

4 Grammatical Rules for English-Farsi Language Pair

As stated earlier, English and Farsi languages have different grammatical structures which results in low quality of translation. Some major challenges of this type, which also affect the translation quality, are discussed in this research. For example, Farsi usually follows SOV pattern in sentences, but this is SVO in English. Also, there may be multiple verbs in a Farsi sentence like English, but there is no clue to find out which verb belongs to which subject and object except the meaning of the sentence. “Ezafe” structure is another feature of Farsi language which makes it challenging in NLP tasks. Ezafe structure is

composed of two or more related words within a phrase which are connected together by Ezafe vowel /e/ or /ye/. Ezafe structure includes:

- A noun before another noun,
- A noun before a possessor,
- A noun before adjectives,
- An adjective before another adjective,
- And combinations of above.

The Ezafe vowel is pronounced but it is not written in Farsi text, thus it raises ambiguities for NLP tasks. One way to reduce such problems is the reordering of words in the source language to simulate the word patterns in the target language. This can be done both by rule-based and data-driven methods where in this research we focus on rule-based reorderings. Regarding to the Farsi language structure compared to English, for English-to-Farsi SMT, two types of reorderings can be applied to the source sentences: Local reorderings which seems appropriate for Ezafe structure and global reorderings which is more suitable for verb reorderings. Global reorderings of verbs puts the verbs in source sentence to the end of the sentence to follow the Farsi structure. This requires the boundaries of clauses especially when there are multiple verbs in a sentence, but there are no obvious marks to determine these points in Farsi sentences. However, an application of hand-crafted rules to reorder the verbs of Farsi sentences in Farsi-to-English SMT is done in [6] by means of conjunctions and punctuations, but using such clues did not lead to notable improvements. Here, we extract some rules for local reorderings of Ezafe structures, which is very common in Farsi, using part of speech tags. These hand-crafted rules are described as follows:

Rule 1: In Farsi, the adjectives in Ezafe structure which describe a noun follow it, whereas in English this order is opposite, i.e. the adjectives precede the noun. For example:

English	a beautiful house and a kind landlord
Reordered English:	a house beautiful and landlord kind

The following rule can be applied to remove this mismatch:

$$\text{JJ [JJ || CC JJ ||, JJ]* [NN ||NNS]} \rightarrow \text{[NN ||NNS] JJ [JJ || CC JJ ||, JJ]*} \qquad \text{Rule (1)}$$

where JJ, CC, NN, and NNS are part of speech tags for adjectives, conjunctions, noun, and plural nouns respectively.

Rule 2: It is also useful to apply reordering when Ezafe occurs in the case of nouns modifying other nouns. In English such relations can be expressed in two ways: 1) using the preposition "of" like "the handle of the door". This pattern matches the Farsi. 2) The order can be changed by removing "of" such as "the door handle". This pattern conflicts Farsi Language. This can be lessened by applying this rule:

$$\text{[NN || NNS]1 [[NN || NNS]2 … [NN || NNS]n} \rightarrow \text{[NN || NNS]n … [NN || NNS]2 [NN || NNS]1} \qquad \text{Rule (2)}$$

Rule 3: Another incompatibility which occurs in Ezafe structure is the placement of pronoun after possessor. For example in English we say "your book", but in Farsi it comes in reverse order "کتاب شما" (ketab-e-shoma).

$$\text{PRO [NN || NNS]} \rightarrow \text{[NN || NNS] PRO} \qquad \text{Rule (3)}$$

where PRO stands for pronoun.

Rule 4: Finally, the order between the noun and its possessor is changed in Farsi. For instance, we say "John's book" in English, but "کتاب جان" (ketabe-e-jaan) in Farsi.

5 Experiments and results

To achieve a reasonable SMT system for English-Farsi, we focus on the bottlenecks of the Farsi language, i.e. limited data resource, text normalization, and grammatical structure of it. To overcome these problems, we gather a large corpus. The statistics of all corpora are shown in Table 2. Then to measure the quality of each of these corpora, we did an experiment. In the following experiments all of the conditions except

the training corpora are the same. These conditions includes language model, tuning set, testing set and translation parameters. Table 3 shows the statistics of the test and tuning sets with four Farsi references and Table 4 demonstrates the quality of each corpus based on BLEU measure:

Test/Tune		Line Number	Singleton	Running Words	Lexicon
Test set	English	418	1945	10981	3144
	Farsi 1	418	1642	12208	2888
	Farsi 2	418	1555	13266	2913
	Farsi 3	418	1366	13021	2673
	Farsi 4	418	1529	12738	2827
Tune set	English	400	2052	10848	3204
	Farsi 1	400	1881	11759	3095
	Farsi 2	400	1825	13235	3136
	Farsi 3	400	1558	12911	2849
	Farsi 4	400	1716	12397	3003

Table 3. Statistics of multi reference test and tuning sets

Corpus	BLEU on Test Set	BLEU on Tuning Set
Central Asia	24.82	24.52
News	27.70	29.76
Misc	20.72	22.61
Ted	14.74	18.23
Verbmobil	4.62	5.68
Existing corpus (Pen, Elra)	7.66	8.34

Table 4. Translation quality on generated corpora (BLEU %)

It is obvious that the News corpus which is translated by human has the best quality.

After generating a big corpus by means of automatic aligners and human translators, we offered the first interactive text normalizer for English-Farsi language pair. This is the first text normalizer for this language pair, which can normalize the text interactively. To show the effectiveness of this tool, we performed three experiments using our big corpus, which is the concatenation of all gathered corpora, plus two existing corpora (Table 2), as the training set and the same corpora of Table 3 as test and tuning sets. In the first trial, an SMT system is created without doing any text normalization on training, testing or tuning sets. Afterward, we did another experiment in which these data sets were normalized statically, i.e. normalizing the text using only a fixed normalization table which consists of valid English-Farsi characters. The final experiment related to this part was to generate a SMT system using interactively normalized data sets. Table 5 indicates the efficiency of the proposed text normalizer on the translation system. Three experiments are done. First, we test the translation system without normalizing the texts. Then we use static text normalization. Finally, interactive normalization is used and the results are as below.

Text Normalization	BLEU on Test Set	BLEU on Tuning Set
None	26.73	28.65
Static approach	27.83	28.60
Interactive approach	29.09	31.04

Table 5. Efficiency of interactive text normalizer (BLEU %)

The experiments clarify that while the static normalization improves quality of the translation, the interactive normalization improves it much more efficiently.

Our final set of experiments is related to the hand-crafted rules which are applied in order to weaken the structural dissimilarities between Farsi and English languages. To this end, four rules, described in section 4, are applied on the source language (English) to make its structure similar to Farsi's. To show the effectiveness of these rules, we perform four experiments. In the first experiment, the baseline system with monotone reordering is created without applying rules. Afterward, we apply the manual rules on the datasets and then create three more SMT systems with monotone, distance-based, and lexicalized reorderings. The results of these experiments are shown in Table 6.

Reordering	Manual Rule	BLEU on Test Set	BLEU on Tuning Set
Monotone	No	26.04	28.19
Monotone	Yes	27.50	30.03
Distance-based	Yes	27.90	30.72

Table 6. Effects of manual reordering (BLEU %)

As the results demonstrate, using manual reordering results in a better BLEU on test set compared to the baseline model with no manual rules and monotone reordering. Since the manual rules are local and we did not apply long range reordering rules, the combination of manual rules and distance-based reordering performs better than

manual rules with monotone reordering. Because of the same reason, i.e. the manual rules do not completely cover the structural differences of English-Persian; it does not perform better than the system which uses lexicalized reordering (Table 5).

6 Conclusion And Future work

In this research we try to create and introduce the first open-domain bilingual English-Farsi corpus which is gathered according to the standard approaches. Then a new text tokenizer/normalizer tool is proposed to normalize, tokenize, and tag the English-Farsi corpus and it is especially designed to interactively normalize the Farsi side to remove the character anomalies in Farsi. Finally, some manual rules are offered to improve the translation quality by decreasing the structural differences of the English-Farsi language pair. Future works includes making use of some other aspects of the proposed normalizer, i.e. the detected tags for special words. Also, find some other effective rules to apply global reordering to English verbs and other useful kinds of distortions to match Farsi sentence patterns.

References

[1] R. Nilipour, "Task- Specific Agrammatism In A Farsi- English Bilingual Patient". JOURNAL OF NEUROLINGUISTICS, NO.4, pages 243-253, 1989.

[2] S. Narayanan, S. Ananthakrishnan, R. Belvin, E. Ettaile, S. Ganjavi,. "Transonics: A Speech To Speech System For English-Persian."In the Proceedings of ASRU. U.S., Virgin Islands, pages 670-675, 2003.

[3] N. Bach, M. Eck, P. Charoenpornsawat, T. Köhler, S. Stüker. "The CMU Transtac 2007 Eyes-Free And Hands-Free Two-Way Speech-To-Speech Translation System.",In the Proceedings of IWSLT, Kyoto, Japan, 2007.

[4] E. Ettelaie, S. Gandhe, P. Georgiou, K. Knight, D. Marcu,S. Narayanan, D. Traum, R. Belvin. "Transonics: A Practical Speech-To-Speech Translator or English-Farsi Medical Dialogs." International Committee on Computational Linguistics and the Association for Computational Linguistics, pages 89-92, 2005.

[5] A. Kathol, J. Zheng. "Strategies For Building A Farsi-English SMT System From Limited Resources." In Interspeech '08, pages 2731–2734, 2008.

[6] E.Matusov, S.Köprü. "Improving Reordering In Statistical Machine Translation From Farsi." in AMTA 2010: The Ninth Conference of the Association for Machine Translation in the Americas. Denver, Colorado, USA, 2010.

[7] T. Chung, M. Galley. "Direct Error Rate Minimization For Statistical Machine Translation." Association for Computational Linguistics, pages 468-479,2012.

[8] F. Och, "Minimum Error Rate Training In Statistical." Association for Computational Linguistics. Sapporo, Japan, pages 160-167, 2003.

[9] http://mokk.bme.hu/resources/hunalign

[10] http://research.microsoft.com/en-us/downloads/aafd5dcf-4dcc-49b2-8a22-f7055113e656/

[11] H. Ney, F. Och, S. Vogel. "Statistical Translation Of Spoken Dialogues In The Verbmobil System." In Workshop on Multi-Lingual Speech Communication, pages 69-74, 2000.

[12] M.A. Farajian, "Pen: Parallel English-Persian News Corpus." Proceedings of the 2011th World Congress in Computer Science, Computer Engineering and Applied Computing, 2011.

[13] http://www.elra.info/LRs-Announcements.html

[14] http://ece.ut.ac.ir/NLP/resources.htm

[15] S. Karimi, "Machine Transliteration Of Proper Names Between English And Persian", PhD thesis, 2008.

[16] http://qt.nokia.com

ARNE - A tool for Namend Entity Recognition from Arabic Text

Carolin Shihadeh
DFKI
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
carolin.shihadeh@dfki.de

Günter Neumann
DFKI
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
neumann@dfki.de

Abstract

In this paper, we study the problem of finding named entities in the Arabic text. For this task we present the development of our pipeline software for Arabic named entity recognition (ARNE), which includes tokenization, morphological analysis, Buckwalter transliteration, part of speech tagging and named entity recognition of person, location and organisation named entities. In our first attempt to recognize named entites, we have used a simple, fast and language independent gazetteer lookup approach. In our second attempt, we have used the morphological analysis provided by our pipeline to remove affixes and observed hence an improvement in our performance. The pipeline presented in this paper, can be used in future as a basis for a named entity recognition system that recognized named entites not only using gazetteers, but also making use of morphological information and part of speech tagging.

1 Introduction

Named entity recognition (NER) is a subtask of natural language processing (NLP). It is the process in which named entities are identified and classified in a text (N. A. Chinchor, 1998). NER is important for NLP, as it supports syntactic analysis of texts and is part of larger tasks, for example information extraction, machine translation or question answering. NLP for the Arabic text is relevant, since Arabic is spoken by more than 500 million people all over the world and there is an enormous number of Arabic sites on the web. The Arabic language has different features that make NLP difficult, such as its complex and rich morphology, the orthographic variation and the non-capitalisation of the Arabic text.

This paper presents a linguistic processing pipeline for Arabic language including tokenization, morphological analysis using a system called ElixirFM developed by Smrz (O. Smrz, 2007), Buckwalter transliteration using the Encode Arabic tool, a placeholder for a part of speech tagger and NER for person, location and organisation named entities. The advantage of such a pipeline model is that the output of one element is the input of the next one, which allows using different resources and information for recognizing named entities. As far as we know, many NER systems combine gazetteers with rules, which consider elements of the surrounding context. In our first approach to recognize named entites from the Arabic text, we have decided to use a gazetteer lookup. A gazetteer is a list of known named entities. If a word is an element in that list then it is labelled as a named entity, otherwise not. The decision of using gazetteers has been influenced by the following criteria:

- Simplicity - developing a NER system which is based on a gazetteer lookup approach is simple.
- Speed - fast execution allow processing large corpora within adequate time.
- Multilingualism - the ability of using the same NER system for any other language, by simply exchanging the used gazetteers.

In our second approach to recognize named entities, we have used the morphological analysis provided

by our pipeline to remove affixes such as the conjunction ”wa” and observed therewith an improvement in our performance. The pipeline presented in this paper, can be used in future as a basis for a NER system that recognized named entites not only using gazetteers, but also making use of morphological information and part of speech tagging.

In section 2 of this paper, we describe some related work done on Arabic NER. In section 3, we present our Arabic named entity recognition pipeline software ARNE. In section 4 and 5, ARNE is evaluated and the results are discussed. Finally, in section 6 we give a conclusion and make some suggestions for future work.

2 Related Work

Named entity recognition from Arabic Text has already been studied before. Systems developed in that field can be basically divided into two types: The first type, is based on a handcrafted approach such as the person NER Arabic system PERA and the NER Arabic system NERA, which were developed by Shaalan et al. (2007, 2008). Shaalan et al. used a handcrafted approach in order to create named entity gazetteers and grammars in form of regular expressions, reporting a f-measure of 92,25% resp. 87.5%. Another system that is based on a handcrafted approach, was developed by Elsebai et al. (2009) who used a grammar based approach in which the grammars can be expressed by using an approach called heuristics definition, reporting a f-measure of 89% (Elsebai et al., 2009). Mesfar (2007) used handcrafted syntactic grammars for his Arabic NER system, reporting a f-measure of 87,3% (S. Mesfar, 2007). The second type of systems, is based on a machine learning (ML) approach. Much work on this field was done by Benajiba et al. using different ML approaches such as maximum entropy, conditional random fields and support vector machines, reporting a f-measures of 55.23% - 83.5% (Benajiba et al.). Also, Maloney and Michael Niv used in their system TAGARAB a ML approach, reporting a f-measure of 85.0% (John Maloney and Michael Niv,1998). Nezda et al. used a ML approach to classify 18 different named entity classes, reporting also a f-measure of 85% (Nezda et. al, 2006). During the development of ARNE we have collected information about several named entity recognizers and summarised their most important features in a table. The table can be provided on demand.

3 ARNE System

ARNE (Arabic Named Entity Recognition) is an Arabic NER pipeline system that recognizes person, location and organisation named entities based on a gazetteer lookup approach. In this section of the paper we are going to describe the development of ARNE and explain its architecture. Figure 1 shows the basic architecture. ARNE makes three preprocessing steps before recognizing the named entities: tokenization, Buckwalter transliteration and part of speech tagging. After the preprocessing steps, ARNE performs a named entity recognition, based on a gazetteer lookup approach. In the following four subsections the subtasks of ARNE are introduced.

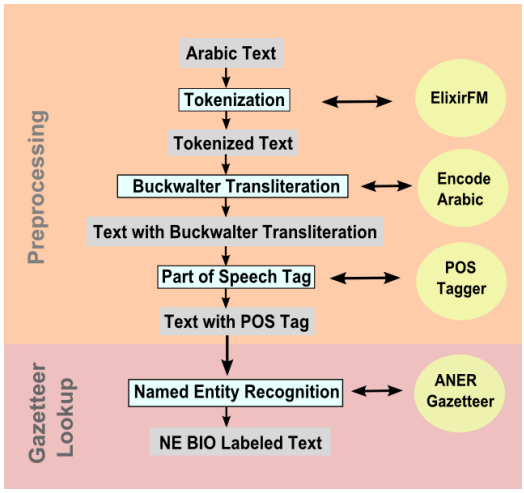


Figure 1: Pipeline architecture of ARNE

3.1 Tokenization

ARNE tokenizes the input text in order to detect the tokens (words, numbers, punctuation marks, special symbols) and sentence boundries. For the Tokenization task in ARNE we have used a system called ElixirFM developed by Smrz (O. Smrz, 2007). The ElixirFM system is able to derive words and inflect them, it can analyse the structure of word forms and

recognize their grammatical function. The input of the tokenization task in ARNE is an Arabic text file. This text file is passed to ElixirFM, which outputs a text that contains the following six columns:

- Column 1: Token
- Column 2: ArabTEX notation which indicates both the pronunciation and the orthography
- Column 3: Buckwalter transliteration of the token depending on its pronunciation in column 2. More details to Buckwalter transliteration follow in 3.2
- Column 4: Morphological analysis
- Column 5: Position of the token in the ElixirFM dictionary
- Column 6: English translation

To represent where a token begins and where it ends, each token is written in one line and between one token and the other there is an empty line. To represent where a sentence ends and where the next sentence begins, two empty lines are left between the last token of the first sentence and the first token of the second sentence. After running ElixirFM on the input text and getting the file that contains the previous mentioned six column, ARNE modifies the output file of ElixirFM and adds a seventh column to it: The Elixir Block number, which is a distinct number that identifies each token in the text and serves there with as a pointer to the information obtained by ElixirFM, as ARNE will not save this information again in the forthcoming steps. The features of ElixirFM (column 2-6) and the Elixir Block number is valuable information, but not needed in our gazetteer lookup approach. We can imagine that this information may be of importance for other approaches made for NER or NLP in general.
Figure 2 is an example of the tokenization task in ARNE when inputting the text:

هاجر ديفيد. الولايات المتحدة قوية

Transliterated as: “hAjr dyfyd. AlwlAyAt AlmtHdp qwyp.” and means: “David emigrated. The United States is powerful.”

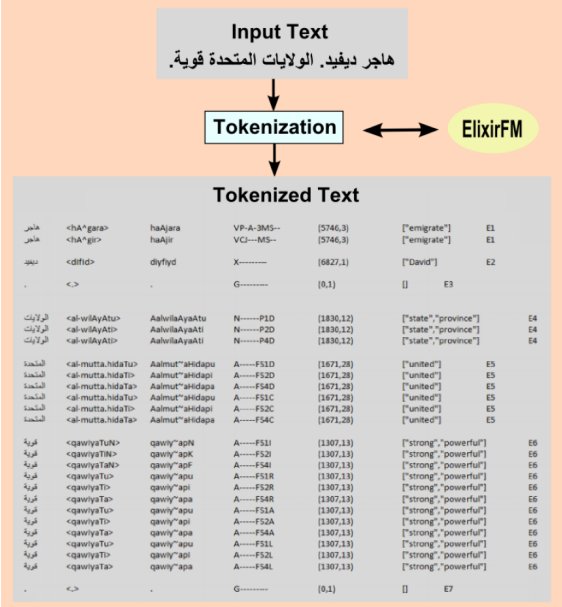


Figure 2: ARNE tokenization of the Text:
هاجر ديفيد. الولايات المتحدة قوية

3.2 Buckwalter Transliteration

We transliterate the Arabic text in order to make it readable for readers who do not have the ability to read the Arabic script but can read the Latin. ARNE uses the Encode Arabic software developed by Tim Buckwalter in order to Buckwalter transliterate the tokens. The input of ARNE in this step is the tokenized text achieved from subsection 3.1. The output is a text that has four columns.

- Column 1: The position of the token in its sentence
- Column 2: The token
- Column 3: The Buckwalter transliteration
- Column 4: The Elixir block number

To represent where one sentence ends and where the other begins we leave an empty line between the sentences and begin numerating the tokens again. As an example for this task, we take the output of Figure 2 as input. The results are illustrated in Figure 3

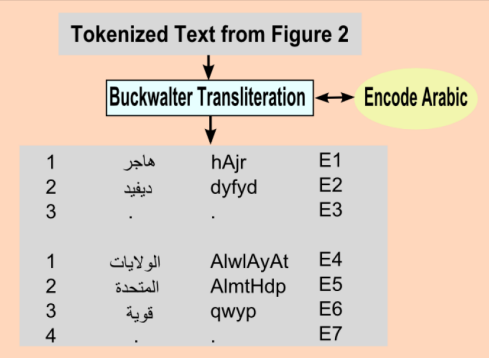


Figure 3: Buckwalter transliteration of the output of Figure 2

3.3 Part of Speech Tagging

This step is responsible for tagging each word with its part of speech. This task was not implemented yet, but we will integrate our own SVM-based tagger which is based on (Gimenez and Marquez, 2004). Initial evaluation on training and testing with the CoNLL 2006 version of the Arabic dependency treebank yields an 95.38% accuracy. Although this experiment has been performed on properly tokenized and transcribed word forms it is very promising. It is no longer a problem for ARNE, that the POS-tagger is not attached to it yet, as we do not need the POS-tag to recognize named entities in our approach. ARNE performs this step, in order to enable integrating a POS-tagger, which may be useful for other NER approaches. The input of this step is the Buckwalter transliterated text from subsection 3.2. The output adds to the input file a column for the POS-tag, which is at the moment the default value "NULL".

3.4 Named Entity Recognition

In this task the person, location and organisation named entities are labelled using the BIO-labelling method. The overall output of the prepossessing step, comes as a file that contains the tokens, their Buckwalter transliteration, possibly a part of speech tag, which is in ARNE at the moment the default value "NULL", and the "Elixir Block Number". For the NER task, ARNE only needs the Buckwalter transliteration of the tokens, as it goes through the text and looks up the token sequences in the ANERgazet gazetteers developed by Benajiba et al. ARNE uses finite automata in order to handle that task, as it has to define the sequences of tokens that are named entities i.e. the language that contains only words (strings) that are named entities contained in the ANERgazet gazetteer. The reason for using finite automata for the language definition task, is that they are fast simulated by computer, do not use much space and can be generated automatically for example using dk.brics.automaton package. In this subsection we are first, going to describe the finite automata of ARNE. Second, we are going to describe the lookup approach that uses those finite automata in order to label the named entities in the text.

3.4.1 ARNE Finite Automata

In order to recognize named entities, ARNE looks up ANERgazet gazetteers which were developed by Benajiba et al. The ANERgazets consists of three gazetteers (Benajiba, 2005 - 2009):

- Person Gazetteer $Gazet_{Pers}$: This gazetteer contains 2309 names, taken from Wikipedia and other websites.
- Location Gazetteer $Gazet_{Loc}$: This gazetteer consists of 1950 names of countries, cities, mountains, rivers and continents found in the Arabic version of Wikipedia.
- Organisation Gazetteer $Gazet_{Org}$: This gazetteer consists of 262 names of football teams, companies and other organisations.

ARNE contains 3 deterministic, minimised finite automata, each automaton recognizes one of the following languages L:

- Person Language:
  $L_{Pers} := \{w \in \sum^* | w \in Gazet_{Pers}\}$
- Location Language:
  $L_{Loc} := \{w \in \sum^* | w \in Gazet_{Loc}\}$
- Organisation Language:
  $L_{Org} := \{w \in \sum^* | w \in Gazet_{Org}\}$

The alphabet consists of the letters that are used for the Buckwalter transliteration i.e. element of the set {A, b, t, v, j, H, x, d, *, r, z, s, $, S, D, T, Z, E, g,

f, q, k, l, m, n, h, w, y, ’, ¿, &, }, —, {, ’, Y, a, u, i, F, N, K, , o, p, ـ , \s}
We have used the dk.brics.automaton java package in order to create for each string in the ANERgazet gazetteers a deterministic finite automaton. After that we merged all those automata to one deterministic finite automaton by creating the power automaton and minimize this automaton using the HOPCROFT algorithm.

3.4.2 ARNE lookup Approach

In this subsection we are going to describe the lookup algorithm used for tagging the tokens with the named entity labels according to the ANERgazet gazetteers, using the BIO-labelling method. A major problem of identifying named entities in text using a gazetter is that named entities are usually multi word entries, especially in Arabic. A simple, but inefficient solution, for extracting the named entities in a text, would be to determine all possible substrings, and match each substring against all gazetters. We will present a more efficient solution, using the morphological analysis provided by our pipeline to remove affixes. The input of ARNE in this step is the POS-tagged text achieved from subsection 3.3. The output is a text that has 6 columns.

- Column 1: The position of the token in its sentence
- Column 2: The token
- Column 3: The Buckwalter transliteration
- Column 4: The POS-tag, at the moment the default value “NULL”
- Column 5: The Elixir block number
- Column 6: The named entity tagb

ARNE looks up strings that have a maximum length of four, because the gazetteers do not contain named entities that consist of more than 4 words. ARNE also assumes that named entities do not cross sentence boundaries, for that reason we handle the named entity labelling task sentence by sentence. The following algorithm, explains how a sentence is labelled using the BIO-labelling method is ARNE.

Lookup Algorithm

INPUT:
Sentence $s := t_1 t_2 ... t_n$
Gazetteer $gazet$: Named entities set

1. **Concatenation:** For practical reasons, concatenate the sentence s with the string NULL NULL NULL

   $s' := t_1 t_2 ... t_n NULLNULLNULL$

2. **Lookup:**

   SET $i := 1$

   WHILE (The end of s' is not reached) **DO:**

   - **CASE$_1$** (Lookup the string $str4 := t_i t_{i+1} t_{i+2} t_{i+3}$) :

     IF ($str4 \in gazet$)
     THEN
     $t_i := B\_NE$
     $t_{i+1} := I\_NE$
     $t_{i+2} := I\_NE$
     $t_{i+3} := I\_NE$
     $i := i + 4$
     GOTO $CASE_1$
     ELSE
     GOTO $CASE_2$

   - **CASE$_2$** (Lookup the string $str3 := t_i t_{i+1} t_{i+2}$) :

     IF ($str3 \in gazet$)
     THEN
     $t_i := B\_NE$
     $t_{i+1} := I\_NE$
     $t_{i+2} := I\_NE$
     $i := i + 3$
     GOTO $CASE_1$
     ELSE
     GOTO $CASE_3$

- **CASE$_3$** (Lookup the string $str2 := t_i t_{i+1}$) **:**

  IF ($str2 \in gazet$)
  THEN
  $t_i := B\_NE$
  $t_{i+1} := I\_NE$
  $i := i + 2$
  GOTO $CASE_1$
  ELSE
  GOTO $CASE_4$

- **CASE$_4$** (Lookup the string $str1 := t_i$) **:**
  IF ($str1 \in gazet$)
  THEN

  $t_i := B\_NE$
  $i := i + 1$
  GOTO $CASE_1$
  ELSE
  $t_i := O$
  GOTO $CASE_4$

END WHILE
OUTPUT: BIO-labelled sentence s

In Figure 4 the task of NE-labelling is illustrated when having the POS-tagged text from Figure 3 as an input.

4 Results

In this section we raise the question of how well ARNE is working in a real application situation. In subsection 4.1 we describe the data used for the evaluation. In subsection 4.2 we present the results of the evaluation.

4.1 Data

For evaluating ARNE, we have used the ANERcorp corpus developed by Benajiba et al. as a goldStandard. The ANERCorp contains more than 150,000 words annotated for the NER task. Since, we use in ARNE the ElixirFM tool for tokenization, we did not have the same tokenization as in the ANERCorp. For the sake of the evaluation, we replaced ElixirFM

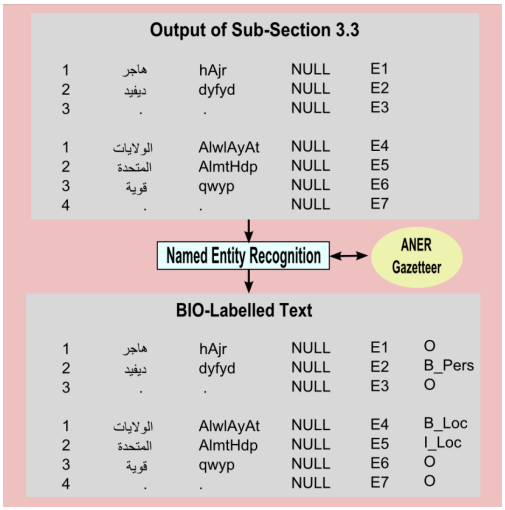


Figure 4: NE-tagged text, when having the POS-tagged text from section 3.3 as an input

in ARNE with a tokenizer that simply tokenizes by "white-space" delimiter and got the same tokenization as in the ANERCorp.

4.2 Evaluation

The basic measures for our evaluation are precision, recall and the f-measure. Table 1 summarizes the results of the evaluation. ARNE achieved a f-measure of 30%, which is basically due the small sizes of the gazetteers currently in use. However, we will indicate, how even in this case morphology can help to improve the quality. In section 5, we discuss the results of the evaluation in more detail and make some suggestions for improvements.

ARNE	Precision	Recall	F1 measure
Person	0.1508	0.2100	0.1756
Location	0.6280	0.4498	0.5242
Organisation	0.3744	0.1397	0.2035
Overall	0.3844	0.2665	0.3010

Table 1: Evaluation

5 Discussion

The advantage of using a gazetteer lookup approach for recognizing named entities is that it is simple, fast and language independent. Achieving a f-measure of 30% in our system ARNE, indicates that

this approach needs improvement. There are several reasons that the f-measure does not reach higher values, for example the size and the quality of the used gazetteers, the rich and complex Arabic morphology which make the tokenization task to a challenge and finally, the ambiguity problem, which is not considered when using a gazetteer lookup approach. The following subsections explain those problems in more detail and show how a higher f-measure can be achieved by solving some of those problems.

5.1 Gazetteer Size and Quality

The quality of the gazetteers is essential, when using a gazetteer lookup approach. A gazetteer should not contain wrong entries. We went manually through the ANERgazet gazetteers and searched for mistakes. In table 2 we list the wrong entries we have found and mention how often they have occured in the ANERCorp corpus which we have used for evaluating our system.

Word	Meaning	Gazetteer	Occurrence
mn	from	PERS	3188
Alywm	today	PERS	149
AlmADy	the past	PERS	128
AlAwl	the first	PERS	40
wA$nTn	Washington	PERS	48
w	and	LOC	217

Table 2: Occurrence of wrong gazetteer entries

We removed the wrong entries from the gazetteers and evaluated again. This small experiments, improved our f-measure from 30% to 32.5%. Table 3 summarizes the results.

ARNE	Precision	Recall	F1 measure
Person	0.2487	0.2095	0.2274
Location	0.6720	0.4587	0.5452
Organisation	0.3744	0.1397	0.2035
Overall	0.4317	0.2693	0.3253

Table 3: Evaluation using modified gazetteers

Not only the quality of the gazetteers play a fundamental role in achieving good results, but also the size of the gazetteers. Many named entities could not be recognized by ARNE, because they are not part of the ANERGazet gazetteers. The ANERGazet gazetteers have been built by Benajiba, who mentions in his thesis that those gazetteers are very small (Benajiba, 2005 -2009). Another problem is, that different writers and typists have a different point of view how things are orthographically correct or permissible and not all computer platforms and keyboards allow the same symbols (Soudi et al., 2007). If a named entity is written in the corpus differently than in the used gazetteers, then ARNE will not be able to recognize that named entity, since the ANERGazet gazetteers do not cover all the possible writing variants of a word. We assume that expanding the used gazetteers would increase the f-measure. But, we should not forget that any person or organisation gazetteer will probably have poor coverage, since new organisations and new person names come into existence every day.

5.2 Ambiguity

Assuming, we succeed to create a gazetteer that has no mistakes and covers all possible named entities then, we will still have the ambiguity problem, since many named entity terms are ambiguous. A NER system without ambiguity resolution, cannot perform robust and accurate NER.

5.3 The Arabic rich and complex morphology

The Arabic language has a complex and rich morphology because it is highly inflectional. One observation we have made was that ARNE could not recognize phrases like

وسوريا

transliterated as “wswryA” which means “and Syria” and is written as “andSyria”. The named entity “Syria” could not be recognized because the gazetteers contain only the named entity “Syria” and not the phrase “andSyria”. We used the morphological information given by ElixirFM to find out whether a phrase contains a conjunction or not and considered this information in our tagging algorithm. Using this morphological information, our f-measure improved from 32.5% to 33.7%. Table 4 summarizes the results.

6 Conclusion and Future Work

We have presented the development of a pipeline software for Arabic named entity recognition (ARNE), which includes tokenization, morphological analysis, Buckwalter transliteration, a place-

ARNE	Precision	Recall	F1 measure
Person	0.2542	0.2159	0.2335
Location	0.6861	0.1400	0.5769
Organisation	0.3676	0.1400	0.2028
Overall	0.4359	0.1653	0.3377

Table 4: Evaluation using morphological information

holder for a part of speech tagger and named entity recognition of person, location and organisation named entities. We have used a gazetteer lookup approach for recognizing named entities from the Arabic text and achieved a f-measure of 30%. Although this low result are basically due the small number of gazetteers, our system provides easy ways of extending it, which is one of our next focus. We have illustrated the boundaries of a gazetteer lookup approach, such as the incapability of creating gazetteers with full coverage and the inability to treat ambiguity. We have demonstrated with some experiments how this performance can be improved, by using for example the morphological information provided by our pipeline.

As future work we intend to integrate a POS-tagger to ARNE, extend the gazetteers, use the POS-tag information and the morphological information provided by ElixirFM to improve the performance and finally, make our lookup algorithm more efficient using parallel programming.

7 Acknowledgments

We wish to thank Dr. Otakar Smrz not only for his system ElxirFM which we have used in our NER system ARNE, but also for the innumerable emails he has written us and the phone calls we had, making us understand the system ElixirFM more deeply and giving us hints how to attach ElixirFM to ARNE. Our thanks goes also to Dr. Yassine Benajiba, who made his gazetteer and corpus available for us and for supporting us to understand his systems ANERsys. We wish also to thank Dr. Khaled Shaalan, Dr. Nizar Habash, Dr. Slim Mesfar, Dr. Hayssam Traboulsi, Dr. Farid Meziane and all people who answered our questions to their papers and made it possible to create the table that summarises work done on Arabic NER. Finally, we would like to thank Alexander Volokh for beta reading.

References

N. A. Chinchor 1998. *Muc-7 named entity task definition (version 3.5), MUC-7.*

K. Shaalan and H. Raza 2007. *Person name entity recognition for arabic, in Semitic 07: Proceedings of the 2007 Workshop on Computational Ap-proaches to Semitic Languages, Morristown, NJ, USA, 2007, Association for Computational Linguistics, pp. 17-24*

K. Shaalan and H. Raza 2009. *NERA: Named Entity Recognition for Arabic. Journal of the American Society for Information Science and Technology archive Volume 60 Issue 8, August 2009 Pages 1652-166 John Wiley and Sons, Inc. New York, NY, USA*

Elsebai, Meziane, Belkredim 2009. *A Rule Based Persons Names Arabic Extraction System. Communications of the IBIMA Volume 11, 2009 ISSN: 1943-7765*

S. Mesfar 2007. *Named entity recognition for arabic using syntactic grammars, in Lecture Notes in Computer Science, Berlin / Heidelberg, , pp. 305-316*

P. R. Yassine Benajiba and J. M. B. Ruiz *Anersys: An arabic named entity recognition system based on maximum entropy*

Yassine Benajiba. *Arabic Named Entity Recognition, PhD thesis* Universidad Politecnica de Valencia.

Yassine Benajiba and P. Rosso *Improving ner in arabic using a morphological tagger.*

P. R. Yassine Benajiba, Mona Diab *Arabic named entity recognition: An svm-based approach.*

John Maloney and Michael Niv *TAGARAB: A Fast, Accurate Arabic Name Recognizer Using High-Precision Morphological Analysis. SRA International Corp. 4300 Fair Lakes Court Fairfax, VA 22033*

2006 Luke Nezda, Andrew Hickl, John Lehmann, and Sarmad Fayyaz *What in the World is a Shahab? Wide Coverage Named Entity Recognition for Arabic. Language Computer Corporation 1701 N. Collins Blvd. Richardson, TX 75080, USA*

O. Smrz 2007. *Functional Arabic Morphology Formal System and Implementa-tion, PhD thesis, CHARLES UNIVERSITY IN PRAGUE*

A. Soudi, A. van den Bosch, and G. Neuman 2007. *Arabic Computational Morphology: Knowledge-based and Empirical Methods, Springer Publishing Company, Incorporated.*

Gimenez, J. and Marquez 2004. *Svmtool: A general pos tagger generator based on support vector machines. In In Proceedings of LREC04, vol. I, pages 43 - 46. Lisbon, Portugal, 2004. (ISBN 2-9517408-1-6).*

Approaches to Arabic Name Transliteration and Matching in the DataFlux Quality Knowledge Base

Brant N. Kay
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
brant.kay@sas.com

Brian C. Rineer
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
brian.rineer@sas.com

Abstract

This paper discusses a hybrid approach to transliterating and matching Arabic names, as implemented in the DataFlux Quality Knowledge Base (QKB), a knowledge base used by data management software systems from SAS Institute, Inc. The approach to transliteration relies on a lexicon of names with their corresponding transliterations as its primary method, and falls back on PERL regular expression rules to transliterate any names that do not exist in the lexicon. Transliteration in the QKB is bi-directional; the technology transliterates Arabic names written in the Arabic script to the Latin script, and transliterates Arabic names written in the Latin script to Arabic. Arabic name matching takes a similar approach and relies on a lexicon of Arabic names and their corresponding transliterations, falling back on phonetic transliteration rules to transliterate names into the Latin script. All names are ultimately rendered in the Latin script before matching takes place. Thus, the technology is capable of matching names across the Arabic and Latin scripts, as well as within the Arabic script or within the Latin script. The goal of the authors of this paper was to build a software system capable of transliterating and matching Arabic names across scripts with an accuracy deemed to be acceptable according to internal software quality standards.

1 Introduction

The challenges inherent to transliterating Arabic names from the Latin script to the Arabic script lie in the fact that there are many seemingly arbitrary ways to spell Arabic names using Latin characters. Halpern (2007) attributes this arbitrariness to the fact that certain Arabic consonant sounds simply do not exist in English, so they are represented in different ways using the Latin script. He also notes that dialectical differences in vowel pronunciation contribute to the variety of Latin spellings. Because there are often several Latin variants of a single Arabic name, it is difficult to successfully transliterate them from Latin to Arabic using a rule-based approach. Take, for example, the name محمد (Latin: *Mohammed*). The single Arabic representation of this name, محمد, can be spelled in several ways using the Latin script. Alternatives include:

Mohamad
Mohamed
Muhamad
Muhamed
Muhammet
Mohammad
Mohammed
Muhammad
Muhammed

Given the variety of spellings in these alternatives, it becomes clear why a lexically-based approach is

necessary to transliterate such names from Latin to Arabic -- rules cannot capture the arbitrary nature of Arabic name orthography as it is rendered using Latin characters. To illustrate this assertion, let's focus on only the two variants *Muhammet* and *Muhammed*. These variants are a minimal pair differing only by their final consonant ('T' or 'D'). The sounds for both 'T' and 'D' are rendered in Arabic as د at the end of the name محمد. One might therefore deduce that a rule can be devised to transform 'T' and 'D' to د at the end of a word. However, mapping both 'T' and 'D' to the Arabic character د is not always appropriate in the word-final context. For instance, the name *Falahat* in Arabic is فلاحت. Mapping the final 'T' in *Falahat* to د would produce فلاحد , which is not a valid transliteration of *Falahat*. To allow for such idiosyncrasies, a list must be built of all known Latin variants of Arabic names, along with their accompanying Arabic transliterations.

There are similar challenges inherent to transliterating Arabic names in the opposite direction -- from the Arabic script to the Latin script. Take, for example, the name *Ruwaida* (Arabic: رويده). The single Latin representation of this name, *Ruwaida*, can be spelled in several ways using the Arabic script. Alternatives include:

رويده
رويدا
رويضه

Focusing specifically on the first two variants, it becomes clear why a rule-based approach will not produce the Latin transliteration *Ruwaida*. رويده and رويدا are a minimal pair differing only by their final character (ه or ا). The sounds for both ه and ا are rendered in Latin as 'A' at the end of the name *Ruwaida*. One might therefore deduce that a rule can be generated to transform ه and ا to 'A' at the end of a word. However, mapping both ه and ا to the Latin character 'A' is not always appropriate in the word-final context. For instance, the name وجيه in Latin is *Wajee*. Mapping the final ه in وجيه to 'A' would produce *Waja*, which is not a valid transliteration for the name وجيه. To allow for this orthographical idiosyncrasy, a list must be built of all known Arabic variants of Arabic names, along with their accompanying Latin transliterations.

There is yet another orthographical complication in Arabic. Arabic is written without short vowels. Halpern (2007) refers to the omission of short vowels as the greatest challenge to achieving accuracy in transliterating Arabic to English. In the absence of information about vowel sounds, there could be several possible transliterations of a single name written in Arabic. Take, for example, فرغل (Latin: *Farghal*). Possible transliterations of this name might include:

Ferghal
Farghal
Firghul
Farghel
Farghil

One must have knowledge of the lexical item فرغل to know that *Farghal* is the proper way to render فرغل using Latin characters. There are no rules that would simply insert short vowels to produce the correct Latin transliteration. To illustrate this assertion we can examine the Arabic name فردوسى, which is properly transliterated to Latin as Firdausi. Both فرغل (Latin: *Farghal*) and فردوسى (Latin: *Firdausi*) begin with the same two Arabic letters ف (Latin: 'F') and ر (Arabic: 'R'). Yet in فرغل we would have to insert an 'A' between these two letters, whereas in فردوسى we would have to insert an 'I' between these two letters to generate each respective Latin transliteration. By definition, no vowel insertion rule can suffice. Knowledge of each lexical item as a whole is necessary for generating the correct Latin transliteration.

The fact that Arabic is not written with short vowels also presents challenges for matching names across scripts when a rule-based approach is employed. Given the absence of vowel information from input in the Arabic script, we must ignore all vowels from input in the Latin script entirely when attempting to compare names across scripts. As a result, certain false matches occur, as seen in the following cluster of names:

Cluster:
خالد
Khaled
خلود
Kholoud

This cluster results from the fact that خالد is transliterated to *Khaled*, whose vowels are then removed via rules to produce the string KHLD.

Likewise, خلود is transliterated to *Kholoud*, whose vowels are then removed via rules to produce the string KHLD. The two Latin input strings *Khaled* and *Kholoud* likewise have their vowels removed via rules, producing the string KHLD in both cases, and all four strings match. Of course, if we consider using placeholders for vowels we could render *Khaled* and *Kholoud* as KH*L*D and KH*L**D, whereby preventing these two Latin renderings from falsely matching. But since Arabic does not contain short vowels, using a placeholder character prevents us from matching Arabic with Latin. There can be no placeholder in Arabic because there are no short vowels to hold on to.

A lexical-based approach would help eliminate this problem of false matches. A list of all known Latin variants and all known Arabic variants of a single name could be mapped to a single canonical Latin representation. خالد and *Khaled* (along with all variants of this name in both scripts) could be mapped to *Khaled*. خلود and *Kholoud* (along with all variants of this name in both scripts) could be mapped to *Kholoud*. The resultant match behavior would produce these two clusters:

Cluster 1:
خالد
Khaled
Cluster 2:
خلود
Kholoud

Hence the problem of false matches can be reduced by using a comprehensive list of names and their variants. A system cannot produce these separate clusters by relying solely on a rule-based approach with a step that removes vowels.

Statistical machine translation-based approaches, such as that described in Hermjakob et. al (2008), have been successful at overcoming many of these challenges. However, the software discussed in this paper relies purely on a deterministic approach to transliteration and matching. The technologies employed in a machine-learning environment were simply not available in the QKB. The QKB is part of a generic system used to analyze and transform data in many languages across different data domains. It is not built to solve any one particular language problem, such as transliterating names between two scripts. Its components are kept simple to enable business users to customize language processing rules to solve a variety of linguistic problems. Therefore the statistical methods required for training on a particular natural language task are not built into its architecture.

2 Method

This section describes the development and testing procedure of the Arabic name transliteration and matching technology, as implemented in the DataFlux Quality Knowledge Base (QKB).

2.1 Arabic to Latin Transliteration

A lexicon of approximately 55,000 Arabic name variants written in the Arabic script, and their accompanying Latin transliterations, was compiled using data acquired from the CJK Dictionary Institute.[1] In addition, an Egyptian subject matter expert manually created a lexicon of approximately 10,000 Arabic name variants written in the Arabic script along with their accompanying preferred Latin transliteration. Since the technology was implemented as part of an Egyptian Arabic software localization project, precedence was given to Egyptian conventions for spelling and spacing within Arabic names written in Latin as the standard for transliterated names. The list of preferred Egyptian transliterations was applied first, followed by the general list of transliterations acquired from the CJK Dictionary Institute. Together these two lexicons served as the primary source for transliteration. Prior to the application of the transliteration lexicons, basic cleansing operations, such as punctuation and diacritics removal, were first applied. As a fall back, rules were designed after the Buckwalter Arabic transliteration scheme[2] to transliterate any names that were not found in either of the two lexicons. Some additional context sensitive rules were added. For example, the ه character transliterates to the A character at a word boundary; elsewhere it becomes H. Three other characters that do not exist in the Buckwalter scheme (ئ , ء, and ؤ) were added as well because they were found in the Egyptian Arabic data that were used to test the system.

[1] http://www.cjk.org/cjk/index.htm
[2] http://open.xerox.com/Services/arabic-morphology/Pages/translit-chart

A sample of 500 full Arabic names was randomly drawn from a population of approximately 9000 full Arabic names written in the Arabic script, taken from a regional banking company's customer database. The 500 names were then transliterated to the Latin script using the QKB. The results were sent to an Egyptian subject matter expert for review. Any transliteration errors were noted in the test results, and the correct transliteration was added to the Egyptian transliteration lexicon. Transliterations were judged as errors if either the lexicon or the fallback rules rendered an unacceptable transliteration according to the subject matter expert. This regression testing process was repeated until the number of errors was deemed to be acceptable according to internal software quality standards.

Example 1: Transliteration via Egyptian transliteration scheme

طارق جعفر ابوالعينين → *Tareq Jafar AboAlEnein*

Example 2: Transliteration via CJK Dictionary Institute lexicon

كاين محرج زيتون → *Kayan Muharrij Zeitoun*

Example 3: Transliteration via PERL regular expression rules

انا نستور مالاخياس → *Ana Nstur Malakhyas*

2.2 Latin to Arabic Transliteration

A lexicon of approximately 863,282 Arabic name variants written in the Latin script, and their accompanying Arabic transliterations, was compiled using data acquired from the CJK Dictionary Institute. Additionally, an Egyptian subject matter expert manually created a lexicon of approximately 10,000 Arabic name variants written in the Latin script along with their accompanying preferred Arabic transliteration. As stated earlier, precedence was given to Egyptian conventions for spelling and spacing, so the list of preferred Egyptian transliterations was applied before the general CJK Dictionary Institute lexicon. Prior to the application of the transliteration lexicons, basic cleansing operations, such as punctuation and diacritics removal, were applied. As a fall back, rules were put in place after the transliteration lists. These rules performed basic letter-for-letter Latin to Arabic transliteration, with some additional context sensitive rules provided by the Egyptian subject matter expert. For example, the Latin characters 'Y' and 'I' are transliterated to the Arabic character ى at word boundaries; elsewhere they become ي. The character 'U' is transliterated to و if it occurs after 'O'; elsewhere it becomes ع.

A sample of 500 full Arabic names was randomly drawn from a population of approximately 8000 full Arabic names written in the Latin script, taken from a regional banking company's customer database. The 500 names were then transliterated to the Arabic script using the QKB. The results were sent to an Egyptian subject matter expert for review. Any transliteration errors were noted in the test results, and the correct transliteration was added to the Egyptian transliteration lexicon. Transliterations were judged as errors if either the CJK Dictionary Institute lexicon or the fallback rules rendered an unacceptable transliteration according to the subject matter expert. This regression testing process was repeated until the number of errors was deemed to be acceptable according to internal software quality standards.

Example 1: Transliteration via Egyptian transliteration scheme

Mohamed Samir AbdElSalam → محمد سمير عبدالسلام

Example 2: Transliteration via CJK Dictionary Institute lexicon

Makhtouf Nesra Abd Elwakel → مقطوف نصراء عبدالوكيل

Example 3: Transliteration via PERL regular expression rules

Anham Enshrah Shaghata → انهام انشراه شاغاته

2.3 Matching

Matching of Arabic names in the QKB is closely related to the Arabic to Latin Transliteration method described above. All names written in the Arabic script are transliterated to Latin in order to match the same, or similar, names across the two scripts.

Prior to applying transliteration lexicons, basic cleansing operations such as punctuation and diacritics removal are applied. As a supplementary step, Arabic name particles in both scripts (ex.

Abdel, Al, El, Abu, ابو ,ال ,عبد) are removed from the input to reduce the input string to a basic canonical representation before final matching. Names in the Arabic script are then transliterated using a lexicon of Arabic names and their Latin counterparts. A second transliteration lexicon, consisting of names in the Arabic script stripped of their particles, is applied. For example, when عبدالرازق (Latin: *AbdelRazek*) is stripped of the particle عبدال (Latin: *Abdel*) in the step above, the name becomes رازق (Latin: *Razek*). The second scheme then transliterates رازق to *Razek*. For any names in the Arabic script that are not in either of the two lexicons, Arabic to Latin phonetic transliteration rules are then applied on a letter-for-letter basis. These rules are similar to the Buckwalter transliterations, but are more simplified in that there are fewer Arabic-to-Latin character mappings. That is, there are more Arabic characters that map to a single Latin character in the phonetic rules than there are in the Buckwalter transliteration scheme. This allows the system to match more names that are similar in pronunciation. After the phonetic transliteration step, all Arabic input is now successfully rendered in the Latin script, and further phonetic reductions (ex. geminate consonant reduction, vowel transformations) take place before final matching.

A sample of approximately 8000 full Arabic names was randomly drawn from a population of approximately 17,000 full Arabic names, half written in Arabic, half in Latin, taken from a regional banking company's customer database. The 8000 names were sent through a cluster analysis test using the matching technology heretofore described. The results were sent to an Egyptian subject matter expert for review. Any false matches or missed matches were noted in the test results, and either the transliteration lexicon or the phonetic transcription rules were updated to yield more accurate match results. This regression testing process was repeated until the number of errors was deemed to be acceptable according to internal software quality standards.

Examples: Clusters of similar names, identified by the matching software system.

Example 1:
فاطمه عباس عبدالرازق
Fatma Abbas Abdel Razek
Fatima Abas Abdel Razik

Example 2:
Ahmed Malawi Abdel-Aaty
احمد معلاوى عبدالعاطى
احمد معلوى عبدالعاطي

3 Results

This section describes the results of the testing procedure of the Arabic name transliteration and matching technology, as implemented in the DataFlux Quality Knowledge Base (QKB).

3.1 Arabic to Latin Transliteration

After twelve iterations of regression testing, the QKB transliterated Arabic names written in the Arabic script to the Latin script with an accuracy of 92%. Testing was halted after twelve iterations because an 8% error rate was deemed acceptable according to internal software quality standards. Once the accuracy reached 92%, returns on further testing iterations became diminished. Customers seeking increased transliteration accuracy for their particular data have the ability to add more names to the existing transliteration schemes. Perfect accuracy was neither necessary nor expected, and thus the product was considered ready to go to market. See above for sample transliterations.

3.2 Latin to Arabic Transliteration

After fourteen iterations of regression testing, the QKB transliterated Arabic names written in the Latin script to the Arabic script with an accuracy of 93.9%. Testing was halted after fourteen iterations because a 6.1% error rate was deemed acceptable according to internal software quality standards. Once the accuracy reached 93.9%, returns on further testing iterations became diminished. Customers seeking increased transliteration accuracy for their particular data have the ability to add more names to the existing transliteration schemes. Perfect accuracy was neither necessary nor expected, and thus the product was considered ready to go to market. See above for sample transliterations.

3.3 Matching

After six iterations of regression testing, the QKB matched names across the Latin and Arabic scripts with an accuracy of 99.6% with respect to false

matches. That is, 0.4% of the matches generated by the QKB were false positives. The accuracy with respect to missed matches was 99.98%; a mere .025% of the data were missed matches; i.e. false negatives. Testing was halted after six iterations because the aforementioned error rates were quite acceptable according to internal software quality standards. See above for sample clusters of similar names.

4 Conclusion

Transliterating and matching Arabic names presents a challenge. Transliterating from Latin to Arabic proves difficult because there are so many Latin variants of a single Arabic name. This variety cannot be readily captured using rules, so a lexicon of Latin to Arabic transliterations must supplement such rules. Transliterating from Arabic to Latin is likewise a challenge for this very same reason. The variety of known Latin transliterations for a single Arabic name means no single transliteration is canonically correct. A list of *preferred* Latin transliterations for the Arabic-speaking country or region in question determines the correct transliteration. Rules schemes such as the Buckwalter Arabic transliteration scheme cannot capture regional orthographic conventions. Finally, the absence of short vowels in the Arabic script means there can be several possible Latin transliterations of a single Arabic name if rules are used. The absence of short vowels in Arabic also accounts for the insufficiency of using rules to match names across scripts. Without vowel information in the Arabic script, we must remove all vowels from the Latin script, and certain false matches occur. The use of a comprehensive lexicon to map all Latin and Arabic variants to a single Latin representation would help solve this problem.

The hybrid approach to transliterating and matching Arabic names, as implemented in the DataFlux Quality Knowledge Base (QKB), performed well in transliterating names across scripts. It should be noted that this paper is reporting on research in progress, as the QKB is continually undergoing updates. As the transliteration lexicons are grown over time, transliteration accuracy will improve. Likewise, any additional contextual rules that may be added to the PERL regular expression rules, and/or the phonetic transliteration rules, will likewise contribute to better transliteration accuracy in both directions. The match results were excellent, most likely due to the significant phonetic reductions, including vowel transformations, which take place after transliteration. On the other hand, we permitted a high tolerance for false positives when evaluating the test results. At the time of development of the QKB's name matching technology, the CJK Dictionary Institute lexicons were not available. In the future, matching will rely less on rules and will leverage the CJK Dictionary Institute lexicons to produce fewer false positives. Further research will involve testing the QKB on more comprehensive data from various sources, followed by subsequent improvements and updates to handle the varying conventions for data formats across different Arabic-speaking regions.

References

Jack Halpern. 2007. The Challenges and Pitfalls of Arabic Romanization and Arabization. In *Proceedings of the Second Workshop on Computational Approaches to Arabic Script-based Languages.* Palo Alta, CA.

U. Hermjakob, K. Knight, and H. Daumé III. 2008. Name Translation in Statistical Machine Translation - Learning when to Transliterate. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 389–397, Columbus, Ohio, June.

Using Arabic Transliteration to Improve Word Alignment from French-Arabic Parallel Corpora

Houda Saadane

LIDILEM - Université Stendhal Grenoble 3
BP 25, 38040 Grenoble Cedex, France
houda.saadane@e.u-grenoble3.fr

Ouafa Benterki, Nasredine Semmar, Christian Fluhr
Institut Supérieur Arabe de Traduction
Rue Tabrizi, 16013 Bir Mourad Raïs, Algérie
obenterki@hotmail.com,
nasredine.semmarn@club-internet.fr
christian.fluhr@gmail.com

Abstract

In this paper, we focus on the use of Arabic transliteration to improve the results of a linguistics-based word alignment approach from parallel text corpora. This approach uses, on the one hand, a bilingual lexicon, named entities, cognates and grammatical tags to align single words, and on the other hand, syntactic dependency relations to align compound words. We have evaluated the word aligner integrating Arabic transliteration using two methods: A manual evaluation of the alignment quality and an evaluation of the impact of this alignment on the translation quality by using the Moses statistical machine translation system. The obtained results show that Arabic transliteration improves the quality of both alignment and translation.

1 Introduction

Transcription consists in replacing each sound or phoneme of a phonological system by a grapheme or a group of graphemes of a writing system, while transliteration consists in replacing each grapheme of a writing system by another grapheme of a group of graphemes of another writing system, regardless of pronunciation. The objective transcription is to reconstruct the original pronunciation using the writing system of the target language and the goal of the transliteration is to represent the original grapheme with the corresponding graphemes of the target languages.

Transcription and transliteration are experiencing significant growth due to the increasingly multilingual Internet and to the exponential needs in the field of cross-language information retrieval (CLIR). This is especially true for finding named entities (names of persons, places, companies, organizations, etc.) but these entities have a plurality of forms, spellings, and transcripts depending on languages and countries. The case of Arabic names illustrates this complex and multifaceted situation. For example, the name of the Libyan leader (Gaddafi), which has a single spelling in Arabic (معمر القذافي) but several pronunciations and accents depending on the dialect, is transcribed into Latin script by over 60 different forms, including: Muammar Qaddafi, Mo'ammar Gadhafi, Muammer Kaddafi, Moammar El Kadhafi,Muammar Gadafi, Moamer El Kazzafi, Mu'ammar al-Qadhdhafi, Mu'amar Qadafi, Muammar Gheddafi, Mu'ammar Al Qathafi, Muʿammar Al-Qaḏâfî…

In this paper, we first outline the theoretical issues and practical difficulties that arise in the transliteration of names and surnames and possible treatments that could resolve these difficulties.

Then, we present, on the one hand, our system for automatic transliteration of Arabic names, and on the other hand, the impact of using transliteration to improve the performance of a word alignment tool.

2 Related Work

The transliteration problem has interested many linguists in different languages, and recently researchers in natural language processing due to the constant development and use of Internet. Many research works have focused on the automatic alignment of transliterations from a multilingual text corpus, in order to enrich bilingual lexicons, which play a vital role in machine translation (MT) and cross-language information retrieval. These include (Al-Onaizan and Knight, 2002) and (Sherif and Kondrak, 2007) who worked on the Arabic-English alignment, (Tao et al., 2006) who work on Arabic, Chinese and English and (Shao and Ng, 2004) who use the information resulted from transliterations based on pronunciation. (Shao and Ng, 2004) combine the obtained information from the translation context and those generated from the Chinese and English transliteration. This technique allows processing some specific infrequent words. We can also find some other systems that assign for a given name only one transliteration such as the generative model for English words written in Japanese (Katakana) to Latin transcription (Knight and Graehl, 1997). This approach was adapted by (Stalls and Knight, 1998) to translate an English word written in Arabic to English. The system of transliteration generation is based on a training dictionary that considers the unknown and unlisted pronunciations within the system. In order to resolve this deficiency, some works have used statistical techniques. This is the case of the transliteration system of the English names to Arabic proposed by (AbdulJaleel and Larkey, 2003). However, this system has several limitations as it uses the computation of the most probable form, supposed to be the correct form but is not always valid in all the Arab countries and dialects. To avoid the pronunciation and dialect's flavor problems, (Alghamdi, 2005) has proposed a transliteration system to translate vowelized Arabic names written in English. This system is based on a dictionary of Arabic names in which the pronunciation is set using vowels added to listed names with an indication of their equivalents in English. Meanwhile, this approach cumulates the disadvantages of the previous techniques: it does considerate the unlisted pronunciations in the dictionary and it is normative as it proposes only one transliteration for a given name. Apparently, the author favored the adoption of a standard transliteration, but this can be only a personal isolated initiative.

Globally, the current works on transcription and transliteration do not reflect their complexity that affects both the oral and writing in two or more linguistic systems in the same time. In fact, transcribing a name from a source linguistic system to another target system is a delicate task which needs some operations requiring management of a set of morphologic, phonetic and semantic properties. These operations are necessary to ensure a robust transliteration process, especially for security, checking identity or information retrieval applications.

However, few studies consider the links between:

- compared phonology and inter-lingual transcription,
- compared graphematic and inter-lingual transcription,
- Arabic dialectology and Latin transliteration systems.

The few studies propose a solution treating partially one of these problematics dedicated to the automatic identification of the speaker origin from its dialect. It is the case of the mentioned studies in (Guidère, 2004) and (Barkat-Defradas et al., 2004).

3 Transliteration of Names Written in Standard Arabic in Latin Characters

The Arabic transcription system includes 28 letters: 25 consonants and 3 vowels that can be short or long according to the word. It contains also some specific morphological and phonological phenomena that must to be taken into account in a transliteration process as the duplication of consonants, sometimes materialized in the Arabic transcription by "shadda", and the repetition of vowels referenced in Arabic by "tanwin". But the

modern Arabic transcription presents the particularity of omitting in general from the texts the indications to the vowels repetition or the short vowels which constitute a source of ambiguity for the transliteration systems.

3.1 Methodology of the Transliterator Construction

We have chosen a "bottom-up" methodology to construct our transliterator. We first start by identifying the existing transliterations for each Arabic letter from the usage norms observed on Internet. This empiric investigation is based on a corpus of texts collected in different languages targeted by the transliterator. It allows to construct a library of graphematic equivalences currently used in the texts transcribed in Latin. In the following table, we present some graphematic equivalences extracted from the used corpus:

Arabic letter	Equivalences in Latin
ء	a
ا	A, a, ä, â, á, ā, e, ê
ب	B, b
ت	T, t
ث	Th, th, t, ţ
غ	Gh, gh, Ğ, ğ, ḡ
ف	F, f, ph
ق	Q, q, C, c, K, k
ك	K, k, C, c
ل	L, l

Table 1: Some graphematic equivalences between Arabic and Latin alphabets.

The study of the corpus allows us to observe that some Arabic letters, without graphematic equivalence in Latin transcription, was transcribed by some Arabic digits in the text written in Latin. This kind of transliteration is particularly used in phone messages (SMS) and the social networks in Europe or Middle East. The following table summarizes these alphanumeric equivalences for the concerned Arabic letters:

Arabic letter	Representation as a number
ء	2
ح	7
خ	7'
ص	9
ض	9'
ط	6
ظ	6'
ع	3
غ	3'
ق	8

Table 2: Transliterations of Arabic letters into numbers.

Hence, by combining these two types of symbolic representations, we can find in the translated texts these equivalences for the usual Arabic names:

Name in Arabic	منى	عدنان	حنان	طارق
Examples of equivalent transcriptions in Latin	Mouna or Mona...	Adnane or 3adnan...	Hanane or 7anan...	Tarek or 6ariq...

Table 3: Examples of Arabic names.

This variation in the use of transliteration is a source of ambiguity when we search information automatically. We can explain this phenomena as follows:

First, for some historical reasons Arab countries were colonized or remanded by some European countries during some periods which were different from one country to another. This occupation has affected the pronunciation, the vocabulary and the transliteration of names of the country's population. Thereby, the influence of the French graphematic and linguistic system is perceptible in the usages of the transliterations in the Maghreb countries, with different intensity from a country to another one. We can see the same thing in the Middle East countries with the English and American influences. Therefore, for political reasons, a common norm does not exist or a unified strategy in the field of transliteration for the Arabic language. This has led every writer or transcriber to use the most dialectal pronunciation

to transcribe the Arabic names. The famous example is that of Laurence of Arabia who, for transcribing the name of the Djeddah city in Saudi Arabia, (جدة), uses 25 times the spelling “Jeddah”, 6 times the spelling “Jidda” and one time the spelling “Jedda” in the same book (in 1926). Laurence of Arabia justified this variation in the transliteration by the following: «we cannot transcribe correctly and with the same manner an Arabic name because the differences between the Arabic and Latin consonants; and the vowel pronunciation which is different from a region to another one» (Alsalman et al., 2007). This is still always true as the different transcriptions of the “Jaddah” cited in Laurence of Arabia are actually used.

Finally, for dialectology reasons, it exists a variety of regional and local dialects in the Arab world. This variety renders impossible finding the same pronunciation for a set of regions or countries. For instance, one of the most frequently used names is the name of the prophet Muhammad (محمد). This name is transcribed in French by Mahomet and has many different pronunciations (transcriptions) like: {Mohamed, Mouhammad, Muhamed, Mhamed, M’Hamed, Muhammad...}. Even when the name is vowelized, it presents many possibilities of transliteration in the texts: {Muhamad, Mouhamad, Mohamad, Mehammad, Mehammade}.

This variation of transliteration according to the dialects is sometimes associated to the use of special characters in some Arab countries or regions. For instance, the following names represent some unconventional forms in Latin transcription: Mu`ammar, Mabrūk, Muṣṭafá, Ismā`īl, Hâdî. All these phenomena require an accurate observation during the process in order to identify the problems and construct efficient rules allowing an automatic process of Arabic names transliteration in real time.

3.2 Description of the Arabic to Latin Translitertor

The module of transliteration of the Arabic script to Latin script is based on finite-state machines (finite-state automata): it consists of states and conditional transitions. Its operation is determined by the nature of the input word: the automaton switches from one state to another according to the outward transitions of the current state and the currently processed letter of the Arabic word. After processing its entire letters, the automaton accepts or rejects the input word. Then the vowels of the input word are removed (if any), and the transliteration is carried out. Finally, the module outputs a sorted list of Arabic names written in Latin characters.

The core of the transliteration system consists of contextual rules. These rules are intended to accurately model the observed forms in the input: is it a "kunya"? A name preceded by an article? Or a first name only?

According to (Guidère, 2006), the name of a person contains several elements in Arabic script. It consists in principle of four main components:

1. The "Kunya" (Particle): typically composed of "Abu" (father of) followed by a name of a child, or of "Umm" (mother of) followed by a name of a child. Example: "Abu Omar" (Father of Omar), "Umm Mohammed" (Mother of Mohammed),
2. The "Ism" (Name): for example, Omar, Ali Mohamed, Khaled Abdallah, etc. It indicates the ethnic or sectarian of the wearer: for example, "Omar" is a typically Sunni name, "Rustam" is a typically Iranian name,"Arslan" is typically Turkish, etc.
3. The "Nasab" (Genealogical affiliation): each name is preceded by "bin" or "Bin/Ben" (Bint/Bent for women). It indicates the exact genealogical descent of the underlying individual. Arabs sometimes go back very far in the indication of the ancestors to avoid confusion among people: ex. Muhammad Salih Bin Abdullah Bin Said Bin, etc.
4. The "Nisba" (suffix of origin): this suffix mainly refers in principle to the tribe or clan in the old genealogy but today it refers specifically to the birthplace of individuals: Maghribi (born in Morocco), Libi (born in Libya), Masri (born in Egypt), Djazaiiri (born in Algeria)…etc. The "Nisba" is always preceded by the

article [Al] and ends with the suffix [i]. It indicates the initial territorial residence of persons, or their nationality.

First, the particles, the part which is not the name itself, are transcribed. Then the transliteration rules are applied to transliterate the names themselves. These transliteration rules are applied in a certain order based on the number of consonants of the name in question and on priority weights. For example, let's consider the name "عبد (Abd) + AL (ال) + Name (رحمن)", the system proceeds as follows:

- Transliteration of the particle عبد (Abd);
- Transliteration of the article ال (Al);
- Concatenating the particle "Abd" and the article "Al" (with a space) and linking them to the name with a hyphen: Abd Al-Rahman (عبد الرحمن);
- Generation of all possible forms of transliteration for these three elements:

Arabic proper name	Transliterations
عبد الرحمن	Abd Al Rahman Abd al-Rahman Abd al Rahman Abd El-Rahman Abd El Rahman Abd el-Rahman Abd el Rahman Abd Ar-Rahman Abd Ar Rahman Abd Ar-Rahman Abd ar-Rahman

Table 4: Some transliteration forms for عبد الرحمن.

An intermediate step allows to overcome some of the very difficult problems of transcription, such as transcription of certain names whose pronunciations change completely for religious or other reasons: this is the case of Moussa translated into Moses, Yusuf into Joseph, Yaakoub into Jacob, Hawa into Eve, etc.

Once the sorted list of transliterated names is generated, the next two tasks are performed:

- Normalization of the list of names in Latin script: This step is to perform some post-processing on the output name in Latin script such as the removal of special characters (diacritics and figures) and changing the first letter into capital (capitalization does not exist in the Arabic script). This notion of capital is retained only in the case of use in databases, but it is not added to the usual search engines, which do not consider the case as relevant;
- Weighting of the output names in Latin script: This step consists in assigning a weight to the rules that were used to generate the list, in order to display the output results sorted from the most likely to the least likely, or vice versa. To achieve this weighting, we use various search engines and the number of occurrences for each generated form of the name: for example, for the Arabic name جمال (jamal), the system generates three different transliterations (Djamel, Jamel, Gamel) and search results frequencies give the following ratios:

Latin transliterated form of the name	Number of occurrences of the name
Djamel	4000000
Jamel	5500000
Gamel	500000

Table 5: Results with Google for the transliterated form of the name جمال.

From the perspective of weighting, this example shows that the Arabic letter (ج) is transcribed, in terms of frequency, mainly by (J), followed by (Dj) and finally by (G).

This procedure has been applied to all the forms of the transliteration of the Arabic characters. It allows establishing a weighted list of equivalences of graphemes that will be used to display the results from the most likely to the least one or vice versa.

4 Using Transliteration to Improve Word Alignment

Word alignment consists of finding correspondences between single words and compound words in a bilingual corpus aligned at

the sentence level. Our word alignment tool uses an existing bilingual lexicon and the following linguistic properties:

- named entities, positions and grammatical categories to align single words,
- syntactic dependency relations to align compound words.

These properties are produced by a linguistic analyzer which is built using a traditional architecture involving separate processing modules:

- A Tokenizer which separates the input text into a list of words.
- A Morphological analyzer which looks up each word in a general full form dictionary. If these words are found, they are associated with their lemmas and all their grammatical tags. For Arabic agglutinated words which are not in the full form dictionary, a clitic stemmer (Larkey et al., 2002) was added to the morphological analyzer. The role of this stemmer is to split agglutinated words into proclitics, simple forms and enclitics.
- A Part-Of-Speech (POS) tagger which searches valid paths through all the possible tags paths using attested trigrams and bigrams sequences. The trigram and bigram sequences are generated from a manually annotated training corpus.
- A Syntactic analyzer which is used to split the list of words into nominal and verbal chain and recognize dependency relations by using a set of syntactic rules. We developed a set of dependency relations to link nouns to other nouns, a noun with a proper noun, a proper noun with the post nominal adjective and a noun with a post nominal adjective. These relations are restricted to the same nominal chain and are used to compute compound words.
- A Named Entity recognizer which uses name triggers such as “Doctor”, “President”, “Government”... to identify named entities (Abuleil and Evens, 2004).

Word alignment using the existing bilingual lexicon consists in extracting for each word of the source sentence the appropriate translation in the bilingual lexicon. The result of this step is a list of lemmas of source words for which one or more translations were found in the bilingual lexicon.

If for a given word no translation is found in the bilingual lexicon and no named entities are present in the source and target sentences, the single-word aligner tries to use grammatical tags of source and target words. This is especially the case when the word to align is surrounded with some words already aligned.

Compound-word alignment consists in establishing correspondences between the compound words of the source sentence and the compound words of the target sentences. First, a syntactic analysis is applied on the source and target sentences in order to extract dependency relations between words and to recognize compound words structures. Then, reformulation rules are applied on these structures to establish correspondences between the compound words of the source sentence and the compound words of the target sentence.

In order to use cognates which are present in the source and target sentences, an additional module was added to the single-word aligner. We consider in our approach pairs of words which share the first four characters as cognates. This step uses the transliteration of proper names and detects for example that the proper name "Jackson" and the transliteration of the Arabic word "جاكسون" (Jackson) are cognates. However, this algorithm does not detect pairs of words such as "Blair" and "bleer" (transliteration of the Arabic word "بلير"). To detect these pairs of words, we defined a similarity based on the number of letters in common rather than simply prefixes. This will also detect proper nouns and numerical expressions. The algorithm for cognates detection was adjusted as follows so that it can select only the words of similar size and with a large number of characters in common regardless of the order of these characters. This algorithm uses the following two parameters:

Words_rate = (Number of characters of the short word) / (Number of characters of the long word)

Cognates_rate = (Number of characters in common) / (Number of characters of the short word)

According to this improvement, two words are cognates if *Words_rate* is greater than 0.8 and *Cognates_rate* is greater than 0.5. These two values are fixed empirically.

This algorithm can certainly identify as cognates the word "blair" and the transliteration "bleer "but it also generates errors as is the case of the couple of words "Muhammad" and the transliteration "mahmoud". To reduce the error rate of this module, we added an additional criterion based on the positions of the two words in the source and target sentences.

Table 6 presents results after running all the steps of word alignment process for single and compound words on the French sentence "*M. Blair a imposé des frais d'inscription élevés à l'université qui ont introduit une sélection par l'argent*." (Mr Blair has imposed high registration fees at the university which introduced a selection by money.) and its Arabic translation " فرض بلير رسوم تسجيل مرتفعة في الجامعة مما ادى الى اختيار الطلاب على قاعدة المال.".

Lemmas of single and compound words of the source language	Lemmas of single and compound words of the target language
Blair (Blair)	بْلِيررَ
imposer (to impose)	فَرَضَ
frais (fees)	رَسْم
inscription (registration)	تَسْجِيل
élevé (high)	مُرْتَفِع
université (university)	جَامِعَة
introduire (introduce)	أدَّى
sélection (selection)	إخْتِيَار
argent (money)	مَال
frais_inscription	رَسْم_تَسْجِيل

Table 6: Result of the alignment of single and compound words.

The word "Blair" was aligned using cognates after transliteration, the words "frais", "élevé" and "introduire" were aligned using grammatical tags and the other single words exist in the bilingual lexicon. The compound word "frais_inscription" was aligned using the reformulation rule *Translation(A.B) = Translation(A).Translation(B)* as follows:

Translation(frais.inscription) = Translation(frais).Translation(inscription) = رَسْم.تَسْجِيل.

5 Experimentation

To evaluate the contribution of the transliteration on the alignment quality of single and compound words, we used two approaches:

- A manual evaluation comparing the results of our word aligner with a reference alignment;
- An automatic evaluation by integrating the results of our word aligner in the training corpus used to extract the translation model of the Moses statistical machine translation system (Koehn et al., 2007).

Because the manual construction of the alignment reference is a difficult and time-consuming task, we conducted a small-scale evaluation based on 283 French-Arabic aligned sentences extracted from the corpus of the ARCADE II campaign. To evaluate the alignment quality, we followed the evaluation framework defined in the shared task on word alignment organized as part of the HLT/NAACL 2003 Workshop on building and using parallel corpora (Mihalcea and Pedersen, 2003). Table 7 summarizes the results of our word aligner in terms of precision and recall. The first line describes the performance of the word aligner when it does not integrate transliteration and the second line mentions its performance when it uses transliteration. As we can see, these results demonstrate that using transliteration improves both precision and recall of word alignment.

Word alignment	Precision	Recall	F-measure
without using transliteration	0.85	0.80	0.82
with the use of transliteration	0.88	0.85	0.86

Table 7: Results of word alignment evaluation.

Certainly, the insufficient size of the corpus used

to evaluate our word aligner does not quantitatively measure the contribution of transliteration but the results clearly indicate an improvement in alignment quality.

The unavailability of a reference alignment of a significant size for single and compound words does not allow us to compare our approach with the state-of-the-art work. That's why we decided to study the impact of the use of the transliteration in word alignment by integrating the results of our word aligner in the training corpus used to extract the translation model of Moses. The initial training corpus is composed of 10000 pairs of French-Arabic sentences extracted from the ARCADE II corpus. We added to this corpus around 10000 pairs of single and compound words corresponding to the results of our word aligner which integrates transliteration on 500 pairs of French-Arabic sentences. We also specified a language model for the target language using the 10800 Arabic sentences of the ARCADE II corpus.

The performance of the Moses statistical machine translation system is evaluated using the BLEU score on a test corpus composed of 250 pairs of sentences. Note that we consider one reference per sentence. In table 8, we report obtained results.

Training corpora	BLEU
without using transliteration	12.50
with the use of transliteration	12.82

Table 8: Translation results with the BLEU score.

This table shows that the inclusion in the training corpus of word alignment results integrating transliteration reports a gain of 0.32 points BLEU.

It is not obvious at this stage to conclude that this gain in BLEU score induces a significant improvement in translation quality given the low value of this score related to the size of used training corpus (only 10000 pairs of sentences for training the translation model and about 10800 sentences to train the target language model). However, we can easily observe that the transliteration improves the performance of the word aligner whatever the used approach for evaluation: manual or automatic.

6 Conclusion

In this article, we described a transliteration system of proper names from Arabic script to Latin script. This system was used in a word alignment process from a French-Arabic corpus. This process is composed of two steps: First, single words are aligned using an existing bilingual lexicon, named entities, positions and grammatical tags, and second, compound words are aligned using the syntactic dependency relations. This process gives satisfactory and encouraging results when the Arabic transliteration is used to align the names present in the source and target sentences. In future work, we plan, on the one hand, to conduct a large evaluation of our word aligner in order to consolidate the obtained results, and on the other hand, to develop strategies to clean word alignment results in order to construct automatically bilingual lexicons from specialized parallel corpora.

References

Abdulmalik Alsalman, Mansour Alghamdi, Khalid Alhuqayl and Salih Alsubay. 2007. Romanization System for Arabic Names. In *Proceedings of the First International Symposium on Computer and Arabic Language ISCAL – 07*, Riyadh, 214-227.

Bonnie Stalls and Kevin Knight. 1998. Translating Names and Technical Terms in Arabic Text. In *Proceedings of the COLING/ACL Workshop on Computational Approches to Semitic Languages*, Montreal, Québec.

Hasnaa Qunaiir. 2001. Romanizing Arabic names. *Journal er-Riyadh*, 12314.

Joseph Dichy. 2009. La polyglossie de l'arabe illustrée par deux corpus. *M. Bozdemir et L.–J. Calvet (EDS), Politiques linguistiques en méditerranée*, Paris: Honoré Champion, 82-102.

Kevin Knight and Jonathan Graehl. 1997. Machine transliteration. In *Proceedings of the 34th ACL Conference*, Madrid, Spain.

Leah S. Larkey, Lisa Ballesteros and Margaret E. Connell. 2002. Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland.

Mansour Alghamdi. 2005. Algorithms for Romanizing Arabic names. *Journal of King Saud University* -

Computer and Information Sciences. Riyadh, 17:01-27.

Mathieu Guidère. 2004. *Le traitement de la parole et la détection des dialectes arabes*. Langues stratégiques et défense nationale, Publications du CREC, Saint-Cyr, 53-75.

Melissa. B. Defradas, Rym Hamdi and François Pellegrino. 2004. De la caractérisation linguistique à l'identification automatique des dialectes arabes, In *Proceedings of the MIDL 2004 Workshop,* Paris, France.

Nasreen AbdulJaleel and Leah. S. Larkey. 2003. Statistical transliteration for English-Arabic Cross Language Information Retrieval. In *Proceedings of the 12th ACM International Conference on Information and Knowledge Management*, New Orleans, LA, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris C. Burch, Marcello Federico, Nicolas Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th ACL Conference, demo session*, Prague, Czech Republic.

Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton, Canada.

Saleem Abuleil and Martha Evens. 2004. Named Entity Recognition and Classification for Text in Arabic. In *Proceedings of the 13th International Conference on Intelligent & Adaptive Systems and Software Engineering*, Nice, France.

Tao Tao, Su-Youn Yoon, Andrew Fister, Richard Sproat and ChengXiang Zhai. 2006. Unsupervised named entity transliteration using temporal and phonetic correlation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06)*, Sydney, Australia.

Tarek Sherif and Grzegorz Kondrak. 2007. Bootstrapping a stochastic transducer for Arabic-English transliteration extraction. In *Proceedings of the 45th ACL Conference*, Prague, Czech Republic. June 2007

Yaser Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In *Proceedings of the 40th ACL Conference*, Philadelphia, USA.

Preprocessing Egyptian Dialect Tweets for Sentiment Mining

Amira Shoukry, Ahmed Rafea
Department of Computer Science and Engineering
The American University in Cairo
Cairo, Egypt
am_magdy@aucegypt.edu, rafea@aucegypt.edu

Abstract

Research done on Arabic sentiment analysis is considered very limited almost in its early steps compared to other languages like English whether at document-level or sentence-level. In this paper, we test the effect of preprocessing (normalization, stemming, and stop words removal) on the performance of an Arabic sentiment analysis system using Arabic tweets from twitter. The sentiment (positive or negative) of the crawled tweets is analyzed to interpret the attitude of the public with regards to topic of interest. Using Twitter as the main source of data reflects the importance of the system for the Middle East region, which mostly speaks Arabic.

Keywords-component; Sentiment; Feature; Tweets; Polarity, Stop-words, Stemming, Normalization

1. Introduction

Sentiment analysis has recently become one of the growing areas of research related to text mining and natural language processing. Due to the increasing availability of online resources and popularity of rich and fast resources for opinion sharing like news, online review sites and personal blogs, several parties such as customers, companies, or even governments started to analyze and explore these opinions. Generally, we can say that determining the writer's attitude regarding some topic or the overall tonality of the text is considered the main task of sentiment analysis. In this paper, we are interested in the effect of the preprocessing stage on the performance of the sentiment classification process for the Arabic language at the sentence level in which the aim is to classify a sentence whether a blog, review, tweet, etc… as holding an overall positive or negative attitude concerning the given topic. It is important to mention that this work is part of a project that will include extracting sentiment topic and other features.

The fields of text mining and information retrieval for the Arabic language had been the interest of many researchers and various studies have been carried in these fields resulting in diverse resources, corpora, and tools available for implementing applications like text classification (Duwairi, 2009) or named entity recognition (Shaalan and Raza, 2009). However, most of the research done in these fields was focused on English texts with very limited research done for other languages such as Arabic (Elhawary and Elfeky, 2010), particularly the Egyptian dialect which is the language of interest for this research. Although Arabic is considered from the top 10 languages mostly used on the Internet based on the ranking carried out by the Internet World State rank in 2010[1] and it is spoken by hundreds of millions of people, there exist very limited annotated resources for sentiment analysis such as labeled corpora, and polarity lexica. This could be considered the main reason which had motivated the generation of an opinion corpus for Arabic in this work.

The majority of the text produced by the social websites is considered to have an unstructured or noisy nature. This is due to the lack of standardization, spelling mistakes, missing punctuation, nonstandard words, repetitions, etc… (Al-Shammari, 2009). That is why the importance of preprocessing this kind of text is attracting the attention these days especially with the presence of several websites producing noisy text. There are mainly three steps in the preprocessing process: 1) normalization, 2) stemming, and 3) stop words removal. Normalization is the process of transforming the text in order to be consistent, thus putting it in a common form. On the other hand, stemming is the process of reducing words to

[1] http://www.internetworldstats.com/

their uninflected base forms. Sometimes the stem is different from the root, but it is useful as usually related words map to the same stem even if this stem is not in itself a valid root. And finally, the stop words removal which is the process of removing those words which are natural language words having very little meaning, such as "في" (in), "علي" (on), "انت" (you),"من" (of), and similar words.

The approaches for sentiment classification are: machine learning (ML) and semantic orientation (SO). The ML approach is a supervised approach where data marked with its class (positive or negative) are used as training data by the classifier implying that a combination of particular features yields a particular class (Morsy and Rafea, 2012) using one of the supervised categorization algorithm like Naïve Bayesian Classifier, Support Vector Machine (SVM), Maximum Entropy, etc… In contrast, the SO approach is mainly an unsupervised approach in which a sentiment lexicon is built with the class of each word is inferred by a number indicating its semantic intensity. Then, all the sentiment words in the sentence are extracted using this lexicon and their polarities are summed up to determine if the sentence has an overall positive or negative sentiment (Morsy and Rafea, 2012). In this study we will be testing the effect of the proposed preprocessing steps on both ML and SO approaches.

The remaining of the paper shows in more details our achieved work in the preprocessing of the Arabic tweets for analyzing and extracting their sentiments. Section II summarizes the work done in the preprocessing stage of most Arabic sentiment mining systems, which is our focus in this study, while section III proposes the system architecture and discusses the system implementation details. Section IV describes the experiments conducted and their results. Finally, Section V talks about the challenges, conclusion and future work.

2. Related Work

Firstly is the normalizing stage which is putting the Arabic text in a consistent form. A normalizer[1] is implemented for doing this job using Ruby. This normalizer performs several tasks such as removing diacritics from the letters, removing 'ء' (Hamza), making both 'ي' and 'ى' change to 'ي'(y), etc…

[1] http://arabtechies.sourceforge.net/projec/ normalization _ruby

Secondly is the stemming stage which is considered one of the most important stages in any Arabic information retrieval or text mining systems. Stemming Arabic terms has proven in several researches that it is not an easy task because of its highly inflected and derivational nature (Larkey. 2007). There are mainly two classes of stemmers for the Arabic language: aggressive stemmers (reducing a given word to its root) and light stemmers (identifying a set of prefixes and suffixes that will be removed). The authors in (Khoja and Garside, 1999) developed an aggressive stemmer which reduces the words to their roots. Their stemmer removes all the punctuation marks, diacritics, numbers, the article "ال" (the), and the inseparable conjunction prefix"و" (and). Additionally, they have built a large prefixes' and suffices' list which is used to check all the input words if they include any of them, and the longest of these is stripped off, if found. Finally, the produced word is compared against a list of patterns and if a match is found, the root is produced. Also, the authors in (Taghva et al., 2005) developed an aggressive stemmer similar to the one described in (Khoja and Garside, 1999) aiming at deriving the root of the word, but they have tried to overcome three issues in that stemmer which they believe are weaknesses in it. The three issues they have identified were: (1) the produced roots are sometimes not related to the original words, (2) the root dictionary which they uses can be difficult to maintain , and (3) the inability of the stemmer to remove affixes that should have been removed. In general, it is noticeable that the problem with aggressive stemmers is that as they reduce the words to their roots, most of the time it results in losing the specific meaning of the original words. This fact has caused this type of stemmers to be poor candidates for systems involving high accuracy in matching between similar words. On the other hand, the authors in (Beltagy and Rafea, 2011) extended one of the existing light stemmers, light10 stemmer, as it is considered to be one of the most accurate available stemmers. Also, they have proposed a set of rules in order to be able to handle broken plurals and transform them to their singular

forms. The approach they have used allowed the stemmer to satisfy accuracy requirements by employing text within a corpus concept to verify whether to carry out such transformation or not. The transformed word is checked to see whether the word resulting from the suggested transformation is present in the corpus or not. So, if a word resulting from applying a transformation rule on an input word (a potential stem), or from removing certain prefixes or suffixes, is found to have appeared in the corpus, then this word is considered as a stem for the input word.

Similarly, the authors in (Nwesri, 2005) compared and proposed a set of techniques for stripping prepositions and conjunctions present at the beginning of a word, after which the result is checked against a lexicon to decide whether that certain prefix should be stripped from the input word or not. And finally, the authors in (Goweder et al., 2004) dealt with the problem of identifying broken plurals and stemming them to their singular forms. In all the experiments they have performed, the input words were first lightly stemmed using the stemmer proposed in (Khoja and Garside, 1999). As a result of observing that this method resulted in very low precision and aiming at improving the results, they have employing one of the machine learning approaches to add restriction rules automatically. But it is noticeable that the best results of all were obtained using a dictionary-based approach.

And finally the stop words removal stage. There is not one definite list of stop words for Arabic. Depending on the type of the application they are implementing, authors use different stop words list. Some authors build lists that consist mainly of the most common and short function words like "في" (in), "من" (of), "علي" (on), etc…[1]. On the other hand, some authors build list that contains the most common words including lexical words like "مثل" (like), "يريد" (want), "يقول" (say), etc…[2].

3. Conceptual Overview

The main aim of this research is to investigate how preprocessing of tweets written in Egyptian dialect could improve the results of sentiment analysis of these tweets. As stated before the preprocessing stage consists mainly of three stages:

3.1 Normalizing the Annotated Tweets

We have used the normalizer [3] as it is very efficient and there is not much work that can be done in this area. Table 1 defines language normalization rules:

Rule	Example
Tashkeel	حدثنا <- حَدَّثَنَا
Tatweel	الله <- اللـــــــــــــــه
Hamza	ء <- ىء or ء or ؤ
Alef	ا <- إ or أ or آ
lamalef	لا <- لإ or لأ or لآ or لا
yeh	ي <- ى or ي
heh	ة <- ه or ة

Table 1. Normalization Rules

3.2 Stemming the Normalized Tweets

Due to the complexity of the Arabic language, several studies with various complexity levels were carried out to address stemming because of its significance in informational retrieval and text mining systems. However, most of these studies were mainly for modern standard Arabic (MSA) and so they can't handle the different dialect specific rules like the Egyptian dialect. For example the word "علشان" (because) if we tried the MSA stemmer the word would become "عش" (hut) since in MSA when a word ends in "ان" it reflects duality, however this word should not have been stemmed originally. The fact which forced us to implement our own customized stemmer. The main objective of the stemmer is to reduce the input word to its shortest possible form without compromising its meaning. That is why we have adopted the light stemming methodology using dialect specific set of prefixes and suffixes because in aggressive stemmers reducing the word to its root can sometimes result in the mapping of too many related terms, each with a unique meaning, to a single root. Moreover, light stemmers are considered very simple to implement and have proven to be highly effective in several information retrieval systems. On the other hand, light stemmers are not applicable of handling some affixes and broken plurals which are very common in the Arabic language (Larkey, 2007). That is why in our

[1] http://www.ranks.nl/resources/stopwords.html
[2] http://arabicstopwords.sourceforge.net
[3] http://arabtechies.sourceforge.net/projec/ normalization _ruby

implemented light stemmer, we have combined some of the rules introduced in (Beltagy and Rafea, 2011), together with a set of rules we have introduced to handle broken plurals for Egyptian dialect which sometimes results in the addition of infixes to a word, as well as handling the removal of certain affixes. In our stemmer's implementation, we have built two lists: one for irregular terms (words that originally start or end by any of the prefixes or suffixes and should not be stemmed) and another one for irregular plurals and their singular forms. These lists are normalized and stemmed. Thus, the input word is first checked against these lists of irregulars if it is present then it won't be stemmed, otherwise the stemming rules will be applied.

The implemented stemmer consists mainly of three stages: 1) prefix removal, 2) suffix removal, and 3) infix removal which is mainly applying the rules for broken plurals. Generally, the prefix removal is the first stage attempted, followed by the suffix removal stage, and finally the infix removal stage. After each stage, the transformed word is checked against the dictionary to determine whether to continue with stemming it, or just stop. Figures 2 and 3 show the sets of prefixes and suffixes proposed for the Egyptian dialect, while figure 4 shows the set rules for handling broken plurals. Most of these rules were inspired from the ones introduced in (Beltagy and Rafea, 2011) with the new ones we propose are highlighted in red.

Prefix	Meaning
ال	The
و ال	And the
بـال	With the
كـال	Like the
فـال	Then the
لـل	For
وبـال	And with the
ولـل	And for
وكـال	And like the
وفـال	And then the
بي	he is
بت	she is
هي	he will
هت	she will
بن	we are
بتت بنت بيت	it is

Prefix	Meaning
و	and
ك	like
ف	then
ل	For or because
ب	With or at

Figure 1: Set of compound and single prefixes with their meanings

Suffix set 1	يات ,ات
Suffix set 2	ها ,ون ,وا ,ين ,ان ,نا ,هم ,ت ,ك ,ي ,ه ,ة ,و

Figure 2: Sets of suffixes

Rule ID	Condition	Rule	Example
R1	Length = 5 & 2nd char != و & 4th character = ئ & 3rd character = ا , & 5th char != ي or ه	Replace 3rd char with a (ي), delete 4th char, and add a (ة) at the end. If no match is found, attempt to find a match without the (ة)	حشائش
R2	Length = 5 & 2nd char = و & 4th character = ء & 3rd character = ا , & 5th char != ي or ه	Remove 2nd char and add a (ة) at the end. If no match is found, attempt to find a match without the (ة)	روائح
R3	Word ends with "ايا" and 2nd character != و	Remove last three chars, and append string (ية)	هدايا
R4	Length = 3 and 2nd character = 3rd character	Remove 3rd character	زمم
R5	Length = 4 and 3rd character = و and last character is not equal to ه	Remove 3rd character (و)	جذور
R6	Length = 5 and 3rd character is an ا	Remove 3rd character (ا)	مراكب
R7	Length = 5 and & 4th character is an ا & 5th character is ء	Remove 4th character (ا) & 5th character is ء and add an (ي) after the 2nd character	وزراء
R8	Length = 5 and 1st character is an ا and last character is a ة	Remove 1st character (ا) and last character (ة) and add an (ا) after the 2nd character	اتربة
R9	Length = 5 and 1st character is an ا and 4th character is an ا and second char != ي	Remove 1st and 4th characters(ا)	اشجار
R10	Length = 4 and 3rd character = و and 2nd character = 4th character	Remove 3rd and 4th characters	سدود
R11	Length = 5 && 2nd character = و and 3rd character is an ا	Remove 2nd character (و)	جوانب

Figure 3: Rules for broken plurals

3.3 Find a List of Egyptian Dialect Stop Words

Given the absence of any stop words list for the Egyptian dialect, we had to build this list from the beginning. The process started by identifying the words in the whole corpus (20,000 tweets) between different frequency ranges as shown in figure 5. The figure shows the number of the words in each frequency range, and it is clear from the graph that there is an inverse relationship between the frequency range and the number of words which complies with Zipf's law (Li, 1992). After that, we started by the first set of words consisting of 11 words which had the highest frequency range to be our list of stop words after removing all the sentiment words "جميل" (beautiful), "بشع" (ugly), etc.. , named entities like "فلول" (Followers), "مصر" (Egypt), "مبارك" (Mubarak), etc…, and verbs like "يحاكم" (Trial), "قتل" (kill), etc…, and tested its effect on the accuracy of the classifier. At the beginning there were drops in performance, means that there might be some important words that should not have been removed, or there some other stop words that still needs to be removed. So we worked on identifying these words manually. Then, this process continued accumulatively by adding lists from the following frequency ranges until we have reached a list of stop words consisting of 128

words that increases the accuracy by almost 1.5%. Figure 6 shows the frequency of each stop.

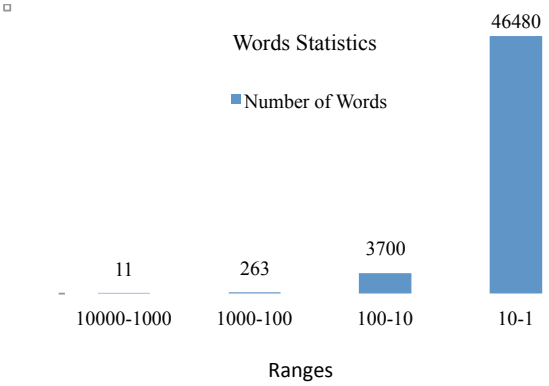


Figure 4. The Number of words in different Frequency Ranges

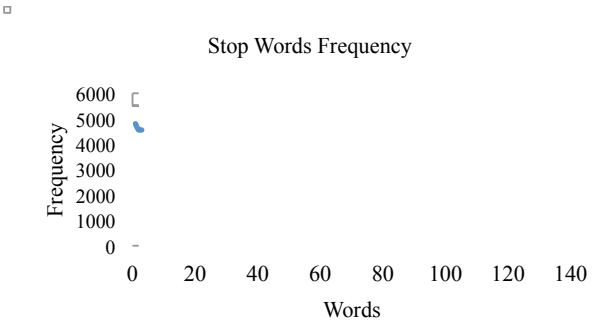


Figure 5. The frequency of each stop word

4. Sentiment Analysis Approaches

The effect of preprocessing on sentiment analysis performance was measured on the two approaches namely ML and SO approaches

4.1 Machine Learning Approach (ML)

This approach uses different feature sets (unigrams, bigrams, and trigrams), together with the Support Vector Machines (SVM) as the machine learning classifier. The preprocessing, features' extraction and the classification are done in three different components, creating the ability to try various arrangements of preprocessing, features and classifiers till reaching the one which yields the highest accuracy.

The methodology used for building the ML classifier consists mainly 5 stages: 1) crawling tweets from twitter to form a corpus, 2) cleaning this created corpus and annotating 1,000 tweets (500 positive tweets and 500 negative tweets), 3) normalizing, stemming and removing the stop words 4) identifying unigrams, bigrams, and trigrams to be used as candidates features in building the feature vectors, 5) using the most known classifier in sentiment classification; SVM. We have used the Weka Suite software (Hall et al., 2009) for the classification process.

4.1.1 Getting Data from Twitter (Arabic Tweets)

Despite the importance of the Arabic language, it is believed to be one of the languages with poor content over the web as very limited number of pages specializes in Arabic reviews. The fact which encouraged us to start using Twitter as the main source for getting vast amounts of data, especially that it provides a search API enabling the search for tweets in the language of interest (Twitter search API, http://search.twitter.com/search.atom?lang=ar& rpp=100&page={0}&q={1}). We were able to crawl more than 20,000 tweets from different news topics. The majority of these crawled tweets were in the Egyptian dialect with small number of tweets in standard Arabic. The size of the corpus was considered one of the main issues as the bigger the size of the training data, the more accurate the classifier will be in classifying any new supplied sentence.

4.1.2 Tweets Cleaning and Annotation

From the 20,000 crawled tweets, 1,000 tweets were annotated (500 positive tweets and 500 negative tweets). For the annotation process, two raters were working on labeling the tweets, and it was observed that they had a high degree of agreement in their classification (over 80%). For those tweets that they labeled differently, a third rater was used to determine its final sentiment. For those annotated tweets, all the user-names, pictures, non-Arabic hash tags, URLs and all non-Arabic words were removed. The tweets selected were chosen based on two assumptions: 1) the sentence represents the opinion of just one author, 2) the sentence holds the author's opinion about only one topic and not sarcastic.

4.1.3 Tweets Pre-Processing

In this stage we just apply the proposed preprocessing tool on the cleaned tweets. Each process is done accumulatively to produce at the

end normalized, stemmed tweets with the stop words removed.

4.1.4 Feature Extraction and Feature Vector

Given that our work is mostly in word/phrase level sentiment analysis, we have chosen to work with unigrams, bigrams and trigrams (Khreisata, 2009). Unigrams are considered the simplest features to extract and they provide good courage for the data, while bigrams and trigrams provide the ability to capture negation or sentiment expression patterns. Therefore, the process starts by extracting all the unigrams, bigrams, and trigrams in the 1000 annotated tweets. Then for each of these candidate features, its frequency in the 20,000 tweets was calculated, creating a dictionary for all the candidate features with their corresponding frequencies. Finally for each Tweet, if any of these candidate features is present in it, then this candidate feature frequency is fetched from the dictionary and it is placed in the feature vector representing this tweet. Therefore, the feature vector built for each tweet used term frequency: ({word1:frequency1, word2:frequency2 …}, "polarity")

4.1.5 Weka Suite Software

For the classification, the Weka Suite Software version 3.6.6 is used as it is a collection of ML algorithms such as NB, SVM, etc… as well as feature selection methods such as IG. Also, various test options exists like configurable number of fold cross validation, test set and percentage split. When the dataset size is large, it is possible to run it directly by inserting the dataset into the program or from the command line.

4.2 Semantic Orientation Approach

The methodology used to build the SO classifier consists mainly of 3 steps: 1) using 600 sentiment annotated tweets (300 positive and 300 negative) to build the sentiment words list, 2) normalizing, stemming and removing the stop words and 3) classifying the remaining 400 tweets (200 positive and 200 negative) as positive or negative using the sentiment word found in the tweet, and building a confusion matrix for the tweets classified as positive and another matrix for the tweets classified as negative to measure the accuracy of classification.

4.2.1 Building the list of Sentiment Words

Given the limited work done for Arabic text in the field of sentiment analysis, especially for the Egyptian dialect, we had first to start by manually building two lists: one for the most occurring positive sentiment words, and one for the most occurring negative sentiment words. Then for each word in these lists a weight is given to it based on its frequency in 300 positive tweets and its frequency in 300 negative tweets.

4.2.2 Tweets Pre-Processing

The same steps (normalizing, stemming, stop words removal) are done in the same order as in the ML approach. Both the tweets and the sentiment words list are processed.

4.2.3 Classifying the Test Set of Tweets

To determine the class of each tweet, a cumulative *score* is calculated using the sentiment words in the tweet to determine its class. For each sentiment words present, its score is added to the total in the following way:

$$score = \sum_{i=1}^{n} (w_{pi} - w_{ni})$$

where w_{pi} is the positive weight of the word, w_{ni} is the negative weight of the word, and they are calculated based on the number of times this word appeared in the positive tweets, and the number of times this word appeared in the negative tweets. The weights assigned to the sentiment words are used to determine how close it is to positive "1" or to negative "-1". The final value of the score (score > 0 or score < 0) determines polarity of the whole tweet. Since, in this stage we are only dealing with two classes building a binary classifier, positive and negative, the neutral class, where either no sentiment words were found or both numbers of positive and negative sentiment words are equal, is not acceptable. Thus for each class a classifier is built determining whether the tweet belongs to its corresponding class, or it belongs to the class named "other". Then, the accuracy, the precision, the recall, and the F-measure of each classifier will be calculated, which will be averaged at the end to reach a final unified accuracy.

5 Experimentation and Evaluation

5.1 ML Results and Discussion

5.1.1 Results

To test the performance of our proposed preprocessing stages, we have applied our 3 stages accumulatively meaning that the normalized tweets will be then stemmed, and finally the stop words will be removed from these stemmed tweets. Four experiments were carried out: 1) using raw tweets, 2) after applying the normalizer, 3) after applying the stemmer, and 4) after removing the stop words; and their results are shown in tables 2, 3, 4 and 5. The SVM classifier was first trained using the frequency of the unigrams only; secondly it was trained using a combination of both unigrams and bigrams with an attempt to capture any negation or sentiment switching phrases; and finally it was trained using a combination of unigrams, bigrams and trigrams to capture any sentiment expression or idioms. The results were as follows using 10-fold validation:

	SVM			
	Accuracy	*Precision*	*Recall*	*F-Measure*
Unigrams	**0.740**	**0.740**	**0.740**	**0.740**
Unigrams + Bigrams	0.739	0.740	0.739	0.739
Unigrams + Bigrams + Trigrams	0.737	0.738	0.737	0.737

Table 2. SVM results using raw tweets

	SVM			
	Accuracy	*Precision*	*Recall*	*F-Measure*
Unigrams	**0.756**	**0.756**	**0.756**	**0.756**
Unigrams + Bigrams	0.754	0.755	0.754	0.754
Unigrams + Bigrams + Trigrams	0.753	0.754	0.753	0.753

Table 3. SVM results using normalized tweets

	SVM			
	Accuracy	*Precision*	*Recall*	*F-Measure*
Unigrams	0.774	0.774	0.774	0.774
Unigrams + Bigrams	0.784	0.784	0.784	0.784
Unigrams + Bigrams + Trigrams	**0.787**	**0.787**	**0.787**	**0.787**

Table 4. SVM results using stemmed tweets (1)

	SVM			
	Accuracy	*Precision*	*Recall*	*F-Measure*
Unigrams	0.738	0.739	0.738	0.738
Unigrams + Bigrams	0.775	0.775	0.775	0.775
Unigrams + Bigrams + Trigrams	**0.779**	**0.779**	**0.779**	**0.779**

Table 5. SVM results using stemmed tweets (2)

	SVM			
	Accuracy	*Precision*	*Recall*	*F-Measure*
Unigrams	0.777	0.777	0.777	0.777
Unigrams + Bigrams	0.788	0.788	0.788	0.788
Unigrams + Bigrams + Trigrams	**0.788**	**0.788**	**0.788**	**0.788**

Table 6. SVM result after stop words removal

Tables 2 shows the results obtained in the classification process for SVM classifier using term frequency scheme respectively before applying any preprocessing, then Table 3-6 show the results obtained after applying each process accumulatively. Tables 4 and 5 show the result of applying two stemmers: 1) our implemented stemmer, and 2) light stemmer [1]. It is important to note that the performance measures of both the

[1] http://pypi.python.org/pypi/Tashaphyne/

positive and the negative classifiers were first calculated using the average of the 10-fold validations, then these measures were averaged to produce the numbers presented in the tables.

5.1.2 Discussion

Comparing the results of SVM, it was clear better results were produced after applying the preprocessing stages. The improvement between the best accuracy results before and after applying preprocessing is almost 4.5%. The same goes with the precision, recall and the F-measure. This behavior was observed in more than one study as preprocessing usually tries to reduce the noise in the text, thus eliminating part of the distortions in the features space. Also an important observation was noticed is that the number of features was reduced dramatically from 6622 features in case of best result using unigrams before applying preprocessing to 4893 features in case of best result using trigrams after applying preprocessing. That is because the more steps we apply from the preprocessing stage, the more related features converge together reducing the problem of features over-fitting and increasing the rate of the learning scheme.

We have tested our implemented stemmer against one of the light stemmers available. Analyzing the results in tables 4 and 5, it is noticeable that our implemented stemmer produces better results because dialect specific issues that we have addressed in our implementation. For example, the word "علشان" and "عشان" both forms of the words are right and they mean "because", in our stemmer we have included them in the irregular list and so they won't be stemmed, however in the light stemmer they will be stemmed to "علش-not a word" and "عش-hut" which are completely two different words.

Regarding the n-gram model, we can note clearly that after applying the stemming, adding the bigram model to the unigram model greatly improves the performance. However, there were not big differences in the performance by adding the trigram model to the combined unigram and bigram model. It should be noted that we have used only the 1000 annotated tweets to build the unigram, bigram and trigrams models, may be using more tweets could result in more unigrams, bigrams and trigrams, thus further improvements in the results.

5.2 SO Results and Discussion

5.2.1 Results

To test the effect of the preprocessing on the SO performance, 3 experiments were carried out one at each stage with the preprocessing applied to both the sentiment words and the tweets. Before carrying the experiments, we have removed stop words as their removal should not have any impact on enhancing the results but their removal will accelerate the classification process. In the first experiment we normalized both the tweets and the sentiment words, and then in the second experiment both were also stemmed. We didn't test the effect of stop words removal on SO performance as there is no intersection between the sentiment words and the stop words, thus removing the stop words won't affect the performance of the SO, it is only the sentiment words which affect it.

	Positive	Negative	Average
Accuracy	0.725	0.653	0.689
Precision	0.768	0.714	0.741
Recall	0.725	0.653	0.689
F-measure	0.746	0.682	0.714

Table 7. SO results using raw tweets

	Positive	Negative	Average
Accuracy	0.728	0.658	0.693
Precision	0.767	0.711	0.739
Recall	0.728	0.658	0.693
F-measure	0.747	0.683	0.715

Table 8. SO results using normalized tweets

	Positive	Negative	Average
Accuracy	0.760	0.758	0.759
Precision	0.761	0.770	0.765
Recall	0.760	0.758	0.759
F-measure	0.760	0.764	0.762

Table 9. SO results using stemmed tweets (1)

	Positive	Negative	Average
Accuracy	0.753	0.755	0.754
Precision	0.758	0.763	0.760
Recall	0.753	0.755	0.754
F-measure	0.755	0.759	0. 757

Table 10. SO results using stemmed tweets (2)

Tables 7-10 calculate the performance results for the classification of the binary classifiers at each stage of the preprocessing. Tables 9 and 10 test the result of applying two stemmers: 1) our implemented stemmer, and 2) light stemmer.

5.2.2 Discussion

Regarding the effect of the preprocessing on the SO performance, we can note that there was an improvement of 7% in the accuracy and the recall, while there was an improvement of 2% in precision and 5% in the F-measure. That is because in SO it is only the form of the sentiment words which affect the performance, thus after preprocessing, the sentiment words in the tweets were almost converted to the same form of the sentiment words in the lists and they were easily extracted. However not all tweets contain sentiment words and even if there exist they represent a very small percentage of the words in the tweet. Hence, building more comprehensive lists of sentiment words could be considered a possible solution to further enhance the performance.

Analyzing the results in tables 9 and 10, it is noticeable that both stemmers produce almost the same results with very minor changes. This behavior is somehow expected as the stemming of most of the sentiment words is expected to be the same because there are less dialect specific sentiment words.

Comparing the results of the positive and the negative binary classifiers, it was clear that the performance of the positive classifier was improving over the performance of the negative classifier until we have applied the stemmer they started to become very close. This behavior reflects the fact that the positive tweets are less noisy than the negative tweets; therefore with minimal preprocessing (just normalizing) it has almost reached the best result.

6. Conclusion and Future Work

In this paper, we have demonstrated the effect of the preprocessing on enhancing the sentiment classification of 1000 Arabic tweets (positive or negative) written in Egyptian dialect from Twitter. As a first step, we believe that the results obtained are very promising. We have used two stemmers (our implemented stemmer and light stemmer) for the aim of comparing their performance in both approaches, and it was noticeable that in ML approach our stemmer produced an improvement of 1% over the light stemmer, while in the SO approach our stemmer produced an improvement of 0.5% over the light stemmer due adding Egyptian dialect prefixes, suffixes and rules for broken plurals. In the ML approach, we have applied the feature vectors to the SVM classifier once before applying the preprocessing and once after applying each stage of the preprocessing to test its effect on the system's performance, and at the end we have reached an improvement in the performance of almost 4.5% in all measures. While in the SO approach, we have applied each stage of the preprocessing to both the tweets and our created sentiment words lists, and at the end we have reached an improvement between 2-7% for the different performance measures.

It is important to note that from the possible causes behind the improvement of the ML approach (78.8%) over the SO approach (75.9%) given that the SO depends only on the sentiment words: 1) the tweet originally contains no sentiment words, 2) the sentiment word in the tweet is not present in the lists, 3) the sentiment word even after applying the preprocessing is written in a different form from the one stored in the list. For example, the word "خير - good" and "خيرا - good", in meaning they are the same but here the suffix "ا" present after the stemming makes them two different words. However, this is considered a defect in the normalization program we are using as "ا" is considered a diacritic that should have been removed.

For future work, we believe that our developed stemmer could be further improved by closely monitoring the performance of each applied rule, thus increasing the probability that more related words will be reduced to the same stems. Also our developed stop words list needs to be further investigated as the performance increased by only 0.1% which means that there are some other stop words that still need to be removed. Moreover, we will be trying to include the semantic to build a hybrid approach combining both ML and SO approaches and testing the effect of preprocessing on this hybrid approach. Accordingly, a more comprehensive list of all Egyptian dialect positive and negative sentiment

words needs to be built since there doesn't exist any of them.

Finally, improving the performance of this preprocessing component with all its stages is currently considered our main aim as it is part of a bigger system for determining sentiment of the Arabic tweets, extracting hot topics, and identifying influential bloggers (Shoukry and Rafea, 2012).

Acknowledgment

The authors would like to thank ITIDA for sponsoring this project entitled "Semantic Analysis and Opinion Mining for Arabic Web", and the Egyptian industrial company LINK-Development and its team for their help in developing a tool for collecting and annotating the tweets.

References

Rehab Duwairi, Mohamed N. Al-Refai, and Natheer Khasawneh. 2009. Feature reduction techniques for Arabic text categorization. Journal of the American Society for Information Science and Technology, 60(11), 2347–2352.

Khaled F. Shaalan and Hafsa Raza. 2009. NERA: Named entity recognition for Arabic. Journal of the American Society for Information Science and Technology, 60(8), 1652–1663.

Sara Morsy and Ahmed Rafea. 2012. Improving Document-Level Sentiment Classification Using Contextual Valence Shifters Natural Language Processing and Information Systems Lecture Notes in Computer Science Volume 7337, 2012, pp 253-258

Leah S. Larkey, Lisa Ballesteros, and Margaret E. Connell. 2007. Light stemming for Arabic information retrieval. In Arabic Computational Morphology, A. Soudi, A. van der Bosch, and G. Neumann, Eds. 221–243.

S. Khoja, and R. Garside. 1999. Stemming Arabic text. Tech. rep. Computing Department, Lancaster University, Lancaster, U.K.

Kazem Taghva, Rania Elkhoury, and Jeffery Coombs. 2005. Arabic stemming without a root dictionary. ITCC 1, 152–157.

Samhaa R. El-Beltagy and Ahmed Rafea. 2011. An accuracy-enhanced light stemmer for arabic text. ACM Trans. Speech Lang. Process. 7, 2, Article 2 (Feb. 2011), 22 pages.

Abdusalam Nwesri, S. M. M. Tahaghoghi, and Falk Scholer. 2005. Stemming Arabic conjunctions and prepositions. In Proceedings of the 12th International Symposium on String Processing and Information Retrieval (SPIRE'05). Lecture Notes in Computer Science, vol. 3772, Springer, 206–217.

Abduelbaset Goweder, Massimo Poesio, and Anne De Roeck. 2004. Broken plural detection for arabic information retrieval. In Proceedings of the Annual ACM Conference on Reseurch and Development in Information Retvieval (SIGIR'04).

Abduelbaset Goweder, Massimo Poesio, Anne De Roeck, and Jeff Reynolds. 2004. Identifying broken plurals in unvowelised Arabic text. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Mohamed Elhawary, and Mohamed Elfeky. 2010. Mining Arabic Business Reviews, Google Inc., Mountain View, CA, USA, 2010 IEEE International Conference on Data Mining Workshops.

Bo Pang and Lillian Lee. Thumbs up? Sentiment Classification using Machine Learning Techniques, Department of Computer Science, Cornell University, Shivakumar Vaithyanathan, IBM Alma den Research Center.

Laila Khreisata. 2009. A machine learning approach for Arabic text classification using N-gram frequency statistics, Journal of Informatics, Vol 3, Issue 1, January 2009, Pages 72-77.

Eiman T. Al-Shammari. 2009. A Novel Algorithm for Normalizing Noisy Arabic Text, Computer Science and Information Engineering, WRI World Congress on , vol.4, no., pp.477-482, March 31 2009-April 2 2009 doi: 10.1109/CSIE.2009.952

Amira Shoukry, and Ahmed Rafea. 2012. Sentence-level Arabic sentiment analysis, Collaboration Technologies and Systems (CTS), 2012 International Conference on , vol., no., pp.546-550, 21-25 May 2012 doi: 10.1109/CTS.2012.6261103

Wentian Li. 1992. Random texts exhibit Zipf's-law-like word frequency distribution, Information Theory, IEEE Transactions on , vol.38, no.6, pp.1842-1845, Nov 1992 doi: 10.1109/18.165464

Rescoring N-Best Hypotheses for Arabic Speech Recognition: A Syntax-Mining Approach

Dia AbuZeina
Palestine Polytechnic University, P.O.Box 198, Hebron, Palestine
abuzeina@ppu.edu

Moustafa Elshafei
King Fahd University of Petroleum and Minerals, P.O.Box 405, Saudi Arabia
shafei@mit.edu

Husni Al-Muhtaseb
King Fahd University of Petroleum and Minerals, P.O.Box 5066, Saudi Arabia
muhtaseb@kfupm.edu.sa

Wasfi Al-Khatib
King Fahd University of Petroleum and Minerals, P.O.Box 5066, Saudi Arabia
wasfi@kfupm.edu.sa

Abstract

Improving speech recognition accuracy through linguistic knowledge is a major research area in automatic speech recognition systems. In this paper, we present a syntax-mining approach to rescore N-Best hypotheses for Arabic speech recognition systems. The method depends on a machine learning tool (WEKA-3-6-5) to extract the N-Best syntactic rules of the Baseline tagged transcription corpus which was tagged using Stanford Arabic tagger. The proposed method was tested using the Baseline system that contains a pronunciation dictionary of 17,236 vocabularies (28,682 words and variants) from 7.57 hours pronunciation corpus of modern standard Arabic (MSA) broadcast news. Using Carnegie Mellon University (CMU) PocketSphinx speech recognition engine, the Baseline system achieved a Word Error Rate (WER) of 16.04 % on a test set of 400 utterances (about 0.57 hours) containing 3585 diacritized words. Even though there were enhancements in some tested files, we found that this method does not lead to significant enhancement (for Arabic). Based on this research work, we conclude this paper by introducing a new design for language models to account for longer-distance constrains, instead of a few proceeding words.

1 Introduction

Improving speech recognition accuracy through linguistic knowledge is a major research area in speech recognition (ASR) systems. Three knowledge sources are usually presented in an ASR: acoustic models, a dictionary, and a language model as shown in Figure 1. These independent knowledge sources, also called ASR database, are subject to adapt to fulfill some natural variations that occur in speech signals. Despite that most of the adaptation occurs in the dictionary, a high integration among the ASR components is required to achieve better performance.

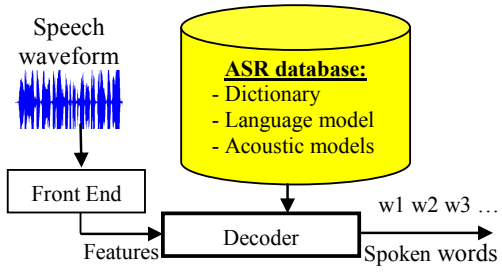


Figure 1. An ASR components

In addition to the pronunciation variation problem, the syntactic structure of the output sentence might be wrong. This problem appears in the form of taking different orders of words and phrases, out of the Arabic correct syntactic structure. Jurafsky and Martin (2009) demonstrated a reason for such phenomenon. They illustrated that variants included in the dictionary may lead to sub-optimal results which can be enhanced using N-Best hypotheses rescoring process. Jurafsky and Martin showed that the Viterbi algorithm is an approximation algorithm. This means that the Viterbi algorithm is biased against words with many pronunciations. The reason for this is that the probabilities' mass is split up among different pronunciations. In Figure 2, the system output, intuitively, is the first hypothesis while the correct output is the second one, which is highlighted. The sentences in Figure 2 are called N-Best hypotheses (also called N-Best list). In this case N is equal 5.

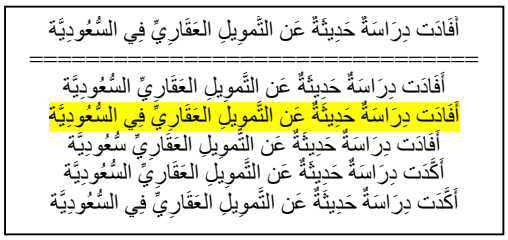


Figure 2. An example of 5-Best hypotheses

To model this problem, the tags of the words will be used as a criterion for rescoring and sorting the N-Best list. We used "language syntax rules" to indicate for the most frequently tags relationships used in the language. The rescored hypotheses are then sorted according to a new weighted scores (acoustic score and syntactic score) to pick the top score hypothesis. Figure 3 shows the idea behind the proposed rescoring model.

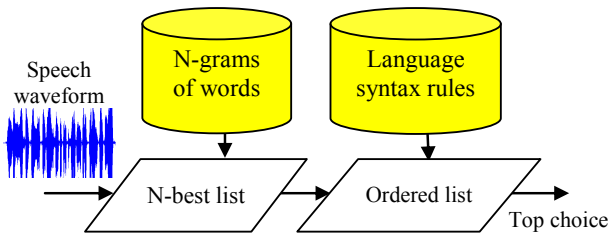


Figure 3. Illustration of rescoring N-Best list

In this work, we utilized the large vocabulary, speaker independent natural Arabic Speech Recognition system developed at King Fahd University of Petroleum and Minerals (KFUPM), based on Carnegie Mellon University (CMU) Pocketsphinx, the state of the art speech recognition engine developed at CMU. Our method is to apply knowledge-based approach for the Arabic sentence structure problem. Certainly, N-Best Arabic syntactic rules are extracted from the tagged Baseline transcription corpus. The extracted rules are then used for rescoring the N-Best hypotheses produced by the ASR decoder.

The paper is organized as follows: In Section 2, we provide a literature review. Sections 3 and 4 introduce data mining approach and the Baseline system, respectively. In section 5, we provide the Arabic phoneme set. Then in Section 6, a description of the Baseline phonetic dictionary is provided. Section 7 describes our methodology followed by Section 8 detailing the testing and evaluation of the proposed method. Then, in section 9, a new design for language models is proposed. Finally, Section 10 presents the conclusion and future work.

2 Literature Review

Using linguistic knowledge to improve speech recognition systems was used by many researchers. Salgado-Garza at al. (2004) demonstrated the usefulness of syntactic trigrams in improving the performance of a speech recognizer for Spanish language. Beutler (2007) demonstrated a method to bridge the gap between statistical language models and elaborate linguistic grammars. He introduced precise linguistic knowledge into a medium vocabulary continuous speech recognizer. His results showed a statistically significant improvement of recognition accuracy on a medium vocabulary continuous speech recognition dictation task. Wang et al. (2002) compared the efficacy of a variety of language models (LMs) for rescoring word graphs and N-Best lists generated by a large vocabulary continuous speech recognizer. These LMs differ based on the level of knowledge used (word, lexical features, syntax) and the type of integration of that knowledge. Xiang et al. (2009) presented advanced techniques that improved the performance of IBM Malay-English speech

translation system significantly. They generated linguistics-driven hierarchical rules to enhance the formal syntax-based translation model.

As Arabic Part of speech (PoS) tagging is essential component in our method, we performed the following literature review. The stochastic method dominates PoS tagging models. Diab et al. (2004) presented a Support Vector Machine (SVM) based approach to automatically tag Arabic text. Al-Shamsi and Guessoum (2006) presented a PoS Tagger for Arabic using a Hidden Markov Model (HMM) approach. El-Hadj et al. (2009) presented an Arabic PoS tagger that uses an HMM model to represent the internal linguistic structure of the Arabic sentence. A corpus composed of old texts extracted from books written in the ninth century AD was created. They presented the characteristics of the Arabic language and the set of tags used. Albared et al. (2010) presented an HMM approach to tackle the PoS tagging problem in Arabic. Finally, the Stanford Natural Language Processing Group developed an Arabic tagger (2011) with an accuracy range between 80% and 96%.

According to the literature review, and to the best of our knowledge, we have not found any research work that employs a machine learning algorithm to distill N-Best syntactic rules to be used for rescoring N-Best hypotheses for large vocabulary continuous speech recognition systems.

3 Data-Mining Approach (WEKA tool)

WEKA is a collection of machine learning algorithms for data mining tasks which represents a process developed to examine large amounts of data routinely collected. Extracting N-Best syntactic rules using WEKA tool is described in Tobias Scheffer (2005). He presented a fast algorithm that finds the *n* best rules which maximize the resulting criterion. The strength of this tool is the ability to find the relationships between tags with no consecutive constraint. For example, if we have a tagged sentence, then it is possible to describe the relations between its tags as follows: if the first word's tag is noun and the sixth word's tag is an adjective, then the ninth word's tag is adverb with certain accuracy. This also could be used for words, i.e. an extracted rule could have *n* words with its relationships and accuracy. Data mining is used in most areas where data are collected such as health, marketing, communications, etc. it worth noting that data mining algorithms require high performance computing machines. For more information about WEKA tool, Please refer to Machine Learning Group at University of Waikato (2011).

4 The Baseline System

Our corpus is based on radio and TV news transcription in the MSA. The audio files were recorded from many Arabic TV news channels, a total of 249 business/economics and sports stories (144 by male speakers, 105 by female speakers), with total duration of 7.57 hours of speech. These audio items contain a reasonable set of vocabulary for development and testing the continuous speech recognition system. The recorded speech was divided into 6146 audio files. The length of wave files varies from 0.8 seconds to 15.1 seconds, with an average file length of 4.43 seconds.

The total words in the corpus are 52,714 words, while the vocabulary is 17,236 words. The transcription of the audio files was first prepared using normal non-vocalized text. Then, an automatic vocalization algorithm was used for fast generation of the Arabic diacritics (short vowels). The algorithm for automatic vocalization is described in detail in Elshafei et al. (2006).The Baseline system WER is reported at 16.04%. Alghamdi et al. (2009) has more details of the pronunciation corpus used in this work.

5 Arabic Phoneme Set

We used the Arabic phoneme set proposed by Ali et al. (2009) which contains (40 phonemes). This phoneme set is chosen based on the previous experience with Arabic text-to-Speech systems (Elshafei 1991, Alghamdi et al. 2004, Elshafei et al. 2002), and the corresponding phoneme set which is successfully incorporated in the CMU English pronunciation dictionary.

6 Arabic Pronunciation Dictionary

Pronunciation dictionaries are essential components of large vocabulary natural language speaker-independent speech recognition systems. For each transcription word, the phonetic dictionary contains its pronunciation in terms of a sequence of phonemes. We used the tool presented

by Ali et al. (2009) to generate a dictionary for the corpus transcription

7 The Proposed Method

Rescoring N-Best hypotheses is the basis of our method. The rescoring process is performed for each hypothesis to find the new score. A hypothesis new score is the total number of the hypothesis' rules that are already found in the language syntax rules (extracted from the tagged transcription corpus). The hypothesis with the maximum matched rules will be considered as the best one. Our method can be described using Figure 4.

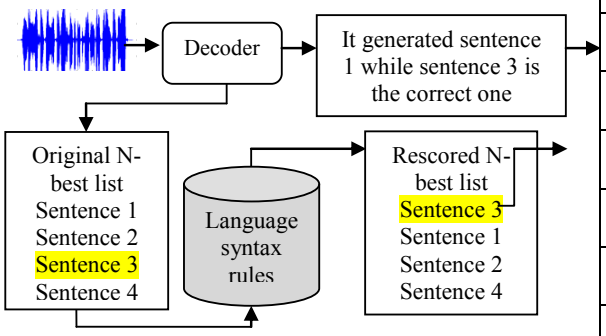


Figure 4. Generation of rescored N-Best list

In Figure 4, suppose that third sentence is the correct sentence that should be returned by the decoder. If the N-Best hypotheses list is rescored using language syntax rules, we expect, hopefully, to get a better result since the final output will be syntactically evaluated. In this case, the hypothesis with maximum number of rules will be chosen since the not-maximum hypothesis is less likely to be the best one. Hence fore, instead of returning the previously top choice (sentence 1) of N-Best list, it will return the top choice of Rescored N-Best list (sentence 3) as shown in Figure 4.

For more clarification, suppose that the two hypotheses of a tested file are as follows:

```
(1)VBD NN NNP DTNNP NN NNP NNP DTJJ DTNN
(2)VBD NN NNS DTNNP JJ NNP NN DTJJ DTNNS
```

Each hypothesis will be evaluated by finding the total number of the hypothesis rules that are already found in the language syntax rules. Suppose that hypothesis number (2) has 4 matching rules while hypothesis number (1) has only 3. In this case, hypothesis number (2) will be chosen as output since it has the maximum matching rules. Since the N-Best hypotheses are sorted according to the acoustic score, if two hypotheses have the same matching rules, the first one will be chosen as it has the highest acoustic score.

Before using WEKA tool, the transcription corpus is tagged using Stanford Arabic tagger which contains 29 tags as shown in Table 1.

#	Tag	Meaning with examples
1	ADJ_NUM	Adjective, Numeric السابع،الرابعة
2	DTJJ	DT + Adjective النفطية،الجديد
3	DTJJR	Adjective, comparative الكبرى،العليا
4	DTNN	DT + Noun, singular or mass المنظمة، العاصمة
5	DTNNP	DT + Proper noun, singular العراق،القاهرة
6	DTNNS	DT + Noun, plural السيارات، الولايات
7	IN	Preposition or subordinating conjunction حرف جر مثل : في حرف مصدري مثل :أنْ
…	…	…
29	UNK	Unknown word

Table 1. Stanford tagging set

Finding language syntax rules is performed using a machine learning tool (WEKA-3-6-5). This tool is called to find N-Best syntactic rules. In our method, we choose to find the best 3000 syntactic rules. For more elaboration, Table 2 shows the first best five rules.

1	TAG4=CD TAG6=DTNN 21 ==> TAG5=IN 21 acc:(0.95635)
2	TAG1=VBD TAG3=DTJJ TAG7=DTNN 21 ==> TAG2=DTNN 21 acc:(0.95635)
3	TAG7=CD TAG8=IN 19 ==> TAG9=DTNN 19 acc:(0.95222)
4	TAG7=CD TAG9=DTNN 19 ==> TAG8=IN 19 acc:(0.95222)

Table 2. First 5-Best syntactic rules of the 3000 rules

Our transcription corpus contains sentences that include up to 30 words. So, our rules have the relationships between tags in the range from 1 to 30. The first rule in Table 2 shows that if the fourth word's tag is a number and the sixth word's tag is a noun, then the fifth word's tag will be preposition with rule accuracy of 95.635%. Rule 2 in Table 2 shows the relationships between distant tags (tag1, tag3, tag7, tag2). As example, the following rule provides the relationships between 6 not-consecutive tags.

```
TAG1=VBD TAG3=DTNN TAG4=DTJJ
TAG5=NN TAG12=NN  ==> TAG2=NN
acc:(0.92298)
```

As we mentioned in section 4 that data mining approach to extract association rules in a large data require a high performance computing (HPC) environment. In our experiments, we found that a desktop computer which contains a single processing chip of 3.2GHz and 2.0 GB of RAM can obtain no more than 530 rules. So, extracting high number of rules in a large corpus requires HPC. We used the HPC at KFUPM which described in HPC Center (2011).

8 Testing and Evaluation

In order to test our proposed method, we split the audio recordings into two sets: a training set and a testing set. The training set contains around 7 hours of audio while the testing set contains the remaining 0.57 hours. We use the CMU language toolkit to build the Baseline language model from the transcription of the fully diacritized text of 7.57 hours of audio. We used the CMU Pocketsphinx to generate the 50-Best hypotheses and, therefore, to test the proposed method. After intensive investigation of our method, we did not find significant enhancement. However, we found enhancements in some tested files as well as new errors introduced in others. Figure 5 and Figure 6 show enhancement in some tested files.

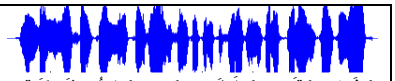
A waveform of a speech sentence with its text form	 هَذَا وَقَد بَلَغَت مَبِيعَاتُ شَرِكَةِ فُورد مُوتُورز فِي الصِّينِ خِلَالَ عَامِ أَلفَينِ وَخَمسَة
As recognized by the Baseline system	هَذَا وَقَد بَلَغَت مَبِيعَاتُ شَرِكَةِ فُورد مُوتُورز التِّسعِينَ خِلَالَ عَامِ أَلفَينِ وَخَمسَة
system	
Found at ➔	Hypothesis # 36
As recognized by the enhanced system	هَذَا وَقَد بَلَغَت مَبِيعَاتُ شَرِكَةِ فُورد مُوتُورز فِي الصِّينَ خِلَالَ عَامِ أَلفَينِ وَخَمسَة

Figure 5. A perfect enhancement in a tested file

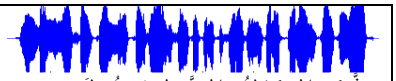
A waveform of a speech sentence with its text form	 حَذَّرَ البَنكُ الدُّوَلِيُّ دُوَلَ الخَلِيجِ العَرَبِيَّة مِن ضَخِّ المَزِيدِ مِن عَائِدَاتِهَا النِّفطِيَّة فِي مَشرُوعَاتٍ
As recognized by the Baseline system	حَذَّرَ البَنكُ الدُّوَلِيُّ دُوَلِ الخَلِيجِ العَرَبِيَّة مِن ضَخم المَزِيدِ مِن عَائِدَاتِهَا النِّفطِيَّة فِي مَشرُوعَاتٍ
Found at ➔	Hypothesis # 50
As recognized by the enhanced system	حَذَّرَ البَنكُ الدُّوَلِيُّ دُوَلِ الخَلِيجِ العَرَبِيَّة مِن ضَخِّ المَزِيدِ مِن عَائِدَاتِهَا النِّفطِيَّة فِي مَشرُوعَاتٍ

Figure 6. A perfect enhancement in a tested file

For the tested file in Figure 5 the best hypothesis was found at position #36, while the hypothesis #50 was found to be best one in Figure 6. The previous two examples show a perfect enhancement where a wrong word is switched to a correct one. The following are two other examples to show partial enhancements in the tested files. Figure 7 found the best choice to be the hypothesis #8, while the hypothesis #4 was found to the best one in Figure 8.

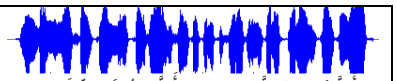
A waveform of a speech sentence with its text form	 وَأَكَّدَ التَّقرِير أَنَّ مُتَوَسِّطَ سِعرِ السَلَّة فِي شَهرِ دِيسَمبَر بَلَغَ ثَمَانِيَةً وَخَمسِينَ دُولارًا وَعَشرَة سِنتَات
As recognized by the Baseline system	وَأَكَّدَ التَّقرِير أَنَّ مُتَوَسِّطَ سِعرِ السَلَّة فِي شَهرِ السَّنِيُورَة بَلَغَ ثَمَانِيَةً وَخَمسِينَ دُولارًا وَعَشرَة سِنتَات
Found at ➔	Hypothesis # 8
As recognized by the enhanced system	وَأَكَّدَ التَّقرِير أَنَّ مُتَوَسِّطَ سِعرِ السَلَّة فِي شَهرِ دِيسَمبَر اللهِ بَلَغَ ثَمَانِيَةً وَخَمسِينَ دُولارًا وَعَشرَة سِنتَات

Figure 7. A partial enhancement in a tested file

A waveform of a speech sentence with its text form	 إنَّ فِـرَقَ الإنـقَـاذ
As recognized by the Baseline system	إنَّ فِـرَقَ الإنـتـرنـت
Found at ➔	Hypothesis # 4
As recognized by the enhanced system	إنَّ فِـرَقَ الإنـقَـاذ الله

Figure 8. A partial enhancement in a tested file

The previous examples show that our method is a promising method to enhance speech recognition accuracy. However, with enhancements in some tested files, we found new errors (i.e. previously correct recognized words) introduced in some tested files as shown in Figure 9.

A waveform of a speech sentence with its text form	 وَذَلِـكَ بِـمُشَـارَكَـةِ عَدَدٍ مِـن رِجَـال أَعمَـال وَمُـستَـثمِـريـنَ سُعُـودِيِّـين
As recognized by the Baseline system	وَذَلِـكَ بِـمُشَـارَكَـةِ عَدَدٍ مِـن رِجَـال أَعمَـال وَمُـستَـثمِـريـنَ سُعُـودِيِّـين
Found at ➔	Hypothesis # 9
As recognized by the enhanced system	وَذَلِـكَ بِـمُشَـارَكَـةِ عَدَدٍ لِـرِجَـالِ أَعمَـال وَمُـستَـثمِـريـنَ سُعُـودِيِّـين

Figure 9. A wrong hypothesis selection example

We also would like to present a case where the N-Best hypotheses already have the correct choice but was not selected after the rescoring process. Figure 10 shows an example.

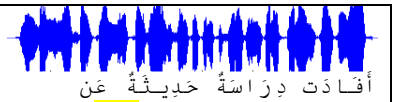
A waveform of a speech sentence with its text form	 أَفَـادَت دِرَاسَةٌ حَدِيـثَـةٌ عَن الـتَّمويـلِ الـعَقَـارِيِّ فِـي الـسُّعُـودِيَّـة
As recognized by the Baseline system	أَفَـادَت دِرَاسَةٌ حَدِيـثَـةٌ عَن الـتَّمويـلِ الـعَقَـارِيِّ الـسُّعُـودِيَّـة
The chosen ➔	Hypothesis # 4
As recognized by the enhanced system	أَفَـادَت دِرَاسَةٌ حَدِيـثَـةٌ عَن الـتَّمويـلِ الـعَقَـارِيِّ سُعُـودِيَّـة
The correct ➔	Hypothesis # 3
Neither Baseline nor enhanced	أَفَـادَت دِرَاسَةٌ حَدِيـثَـةٌ عَن الـتَّمويـلِ الـعَقَـارِيِّ فِـي الـسُّعُـودِيَّـة

Figure 10. Not-selected correct hypothesis example

In our method, part of speech tagging was crucial to support the correctness of the method used. Even though the Stanford tagger which was used in our method has many correct tagged sentences, however, there are many mistakenly tagged sentences. We provide two examples of a correct tagged sentence and a wrong tagged one as shown in Figure 11.

A correct tagged sentence
قـالـت/VBD شركـة/NN أرامـكو/NNP الـسعـوديـة/DTNNP وشركـة/NN دال/NNP كـيـمـيـكـلـز/NNP الأمـريـكـيـة/DTJJ الـيـوم/DTNN
A wrong tagged sentence
وقـال/NN مـتـقـي/JJ إن/NN الـجـمـهـوريـة/DTNN الإسلامـيـة/DTJJ مـصـمـمـة/VN عـلـى/IN أن/NN تـكـون/VBP مـزودا/VN لـلـنـفـط/NN فـعـلا/NN بـالـثـقـة/JJ وجـديـرا/NN

Figure 11. Two examples of tagged sentences

In Figure 11, the highlighted texts were wrongly tagged. So, extracting the language syntax rules using many errors will not be strong enough for rescoring the N-Best hypotheses. This is our justification of our result, enhancement in some tested files and new errors in others.

In addition to the tagger problem, we finalize this section by explaining the effect of diacritics in this research work. Not like English, Arabic sentences are diacritized. Accordingly, the N-Best

hypotheses will be diacritized. Acoustic score also provided for each hypothesis as shows in Figure 12.

```
9106- الَّتِي تَعتَمِدُ عَلَى الغَاز فِي السُّعُودِيَّة
9179- الَّتِي تَعتَمِدُ عَلَى الغَاز فِي السُّعُودِيَّة
9320- الَّتِي تَعتَمِدُ عَلَى الغَاز فِي السُّعُودِيَّة
9130- الَّتِي تَعتَمِدُ عَلَى الغَاز فِي السُّعُودِيَّة
9203- الَّتِي تَعتَمِدُ عَلَى الغَاز فِي السُّعُودِيَّة
9344- الَّتِي تَعتَمِدُ عَلَى الغَاز فِي السُّعُودِيَّة
9564- الَّتِي تَعتَمِدُ عَلَى الغَاز السُّعُودِيَّة
9588- الَّتِي تَعتَمِدُ عَلَى الغَاز السُّعُودِيَّة
9609- الَّتِي تَعتَمِدُ عَلَى الغَاز السُّعُودِيَّة
9633- الَّتِي تَعتَمِدُ عَلَى الغَاز السُّعُودِيَّة
9655- الَّتِي تَعتَمِدُ عَلَى الغَاز السُّعُودِيَّة
9679- الَّتِي تَعتَمِدُ عَلَى الغَاز السُّعُودِيَّة
9756- الَّتِي تَعتَمِدُ عَلَى الغَاز السُّعُودِيَّة
9780- الَّتِي تَعتَمِدُ عَلَى الغَاز السُّعُودِيَّة
9909- الَّتِي تَعتَمِدُ عَلَى لِمَصفَى السُّعُودِيَّة
```

Figure 12. 10-Best list of a tested file.

It is noted that the N-best hypotheses produced by the ASR system are diacritized, which results in many hypotheses that differ only in the diacritics, thus reducing the variety of hypotheses that are included in the N-best list for any value of N. The highlighted hypotheses in Figure 12 are examples. This same-tags case prevents the diversity that should be presented in the N-Best hypotheses. One case, among 300-Best hypothesis, we found 16 different hypotheses, (i.e. at words level). As the acoustic scores are sorted in decreasing order, the problem showed up when, as example, finding the first 50 hypotheses with same words and different diacritics. So, instead of searching among first different hypotheses like English, the search will be away from the high score results, therefore, reducing the accuracy.

9 New Designs for Language Models

Even though our method does not increase the Baseline accuracy, it introduces a new design for language models. We propose to relax the constraint of having consecutive few words which usually used to build language models. Cao et al. (2006) demonstrated that many manually identified relationships can be hardly extracted automatically from corpora. This is why they used hand-crafted thesauri (such as WordNet) and co-occurrence relationships for limited relations related to nouns (synonym, hypernym and hyponym). Ruiz-Casado et al. (2007) describes an automatic approach to identify lexical patterns that represent semantic relationships between concepts in an on-line encyclopedia. They have found general patterns for the hyperonymy, hyponymy, holonymy and meronymy relations. Figure 13 shows our proposed framework. It shows that instead of finding words relations based on specific types, we propose to find words' relations with no restrictions (i.e. in general)

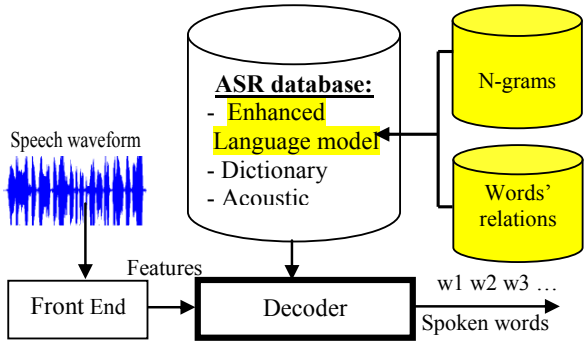


Figure 13. A proposed framework for language models

Figure 13 shows that instead of building the language models based on few consecutive words, the language models could account for longer-distance constrains which we called Enhanced language model. The longer-distance relations have no constraints regard the number of words (such as two or three) or type (such as synonyms). As we mentioned in section 8 (the proposed method) that WEKA tool can extract the relations of many tags. In the same way, we propose to use WEKA to extract the relationships between different words within the same sentence. There are no restrictions of the numbers of words, as the current language models which deal with 3 consecutive words maximum. WEKA tool can generate N-Best rules which can be used as a complement module of the s tandard language models. In this case, instead on having one module, two modules will be used in computation the words consecutive score. For example, the following cases illustrate how to utilize WEKA tool to extract words' relationships. So, as the rule:

```
TAG1=VBD TAG3=DTNN TAG4=DTJJ
TAG5=NN TAG12=NN ==> TAG2=NN
```

We can extract a similar rule but directly with words as follows:

word1=حددت word3=الحج
word4=السعودية word5=معيار
word12=المقبل ==> word2=وزارة

In this case, 6 words can contribute to find the best sentence which is better than n-grams which require the words to be executives and usually built using (2-3) words.

10 Conclusion and Future Work

In this paper, we conclude that N-Best rescoring for Arabic speech recognition (using Arabic data-driven syntax) does not provide significant enhancement. However, more investigation can be performed with a high accurate part of speech tagging model.

As future work, we recommend to utilize linguistic knowledge at the decoder level, i.e. before releasing the decoder output. We also recommend to do further research on Arabic part of speech tagging, especially for diacritized text.

Acknowledgments

This work is supported by Saudi Arabia Government research grant NSTP # (08-INF100-4). The authors would like also to thank King Fahd University of Petroleum and Minerals for its support of this research work.

References

Bing Xiang, Bowen Zhou and Martin Cmejrek. 2009. Advances in syntax-based Malay-English speech translation. Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE Computer Society: 4801-4804.

Dan Jurafsky and Martin J. 2009. Speech and Language Processing, second edition, Pearson.

Fatma Al-Shamsi and Ahmed Guessoum. 2006. A Hidden Markov Model –Based PS Tagger for Arabic, CiteSeerX.

Guihong Cao, Jian-Yun Nie, and Jing Bai. 2005. Integrating word relationships into language models. Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. Salvador, Brazil, ACM: 298-305.

High Performance Computing (HPC) Center, 2011. http://hpc.kfupm.edu.sa/Home.htm

Luis R. Salgado-Garza, Richard M. Stern and Juan A. Nolazco F. 2004. N-Best List Rescoring Using Syntactic Trigrams ,MICAI 2004: Advances in Artificial Intelligence.

Machine Learning Group at University of Waikato, 2011.http://www.cs.waikato.ac.nz/ml/WEKA/

Mansour Alghamdi , Moustafa Elshafei , and Husni Almuhtasib . 2009. Arabic broadcast news transcription system, International journal of speech and technology, 10: 183–195

Mansour Alghamdi , Husni Almuhtasib, and Moustafa Elshafei . 2004. Arabic Phonological Rules, King Saud University Journal: Computer Sciences and Information. Vol. 16, pp. 1-25

Maria Ruiz-casado , Enrique Alfonseca , and Pablo Castells. 2007. Automatising the learning of lexical patterns: an application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. Data Knowledge and Engineering, 61 3 , pp. 484–499.

Mohammed Albared, Nazlia Omar, Mohd.Aziz, and Mohd Ahmad Nazri. 2010. Automatic part of speec tagging for Arabic: an experiment using Bigram hidden Markov model, RSKT'10 Proceedings of the 5th international conference on Rough set and knowledge technology

Mohamed Ali, Moustafa Elshafei, Mansour Alghamdi, Husni Almuhtaseb and Atef Alnajjar, 2009. Arabic Phonetic Dictionaries for Speech Recognition. Journal of Information Technology Research, Volume 2, Issue 4, pp. 67-80.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of Arabic text: from raw text to base phrase chunks, 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference.

Moustafa Elshafei. 1991. Toward an Arabic Text-to-Speech System, The Arabian Journal of Science and Engineering, Vol. 16, No. 4B, pp.565-583.

Moustafa Elshafei , Husni Almuhtasib and Mansour Alghamdi. 2002. Techniques for High Quality Text-to-speech, Information Science, 140 (3-4) 255-267.

Moustafa Elshafei., Husni Al-Muhtaseb, and Mansour Alghamdi. 2006 . Machine generation of Arabic diacritical marks. In Proceedings of the 2006 international conference on machine learning; models, technologies, and applications (MLMTA'06), USA.

René Beutler. 2007. Improving Speech Recognition through Linguistic Knowledge, Doctoral Dissertation ,ETH Zurich.

Stanford Log-linear Part-Of-Speech Tagger, 2011. http://nlp.stanford.edu/software/tagger.shtml

Tobias Scheffer . 2005. Finding association rules that trade support optimally against confidence. Intell. Data Anal. 9(4): 381-3

Wen Wang, Yang Liu, and Mary P. Harper. 2002. Rescoring effectiveness of language models using different levels of knowledge and their integration. Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on.

Yahya El Hadj, Imad Abdulrahman Al-Sughayeir and Abdullah Mahdi Al-Ansari. 2009. Arabic Part-Of-Speech Tagging using the Sentence Structure, Proceedings of the Second International Conference on Arabic Language Resources and Tools, The MEDAR Consortium.

Morphological Segmentation and Part of Speech Tagging for Religious Arabic

Emad Mohamed
Carnegie Mellon University Qatar
emohamed@qatar.cmu.edu

Abstract

We annotate a small corpus of religious Arabic with morphological segmentation boundaries and fine-grained segment-based part of speech tags. Experiments on both segmentation and POS tagging show that the religious corpus-trained segmenter and POS tagger outperform the Arabic Treebak-trained ones although the latter is 21 times as big , which shows the need for building religious Arabic linguistic resources. The small corpus we annotate improves segmentation accuracy by 5% absolute (from 90.84% to 95.70%), and POS tagging by 9% absolute (from 82.22% to 91.26) when using gold standard segmentation, and by 9.6% absolute (from 78.62% to 88.22) when using automatic segmentation.

1 Introduction

Traditional religious Arabic is the language variety used in pre-Modern texts dealing with the Quran, prophetic traditions, and the various books on Islamic law, Quran interpretation, Islamic philosophy and many other fields. It has more or less the same structure as Modern Standard Arabic but contains lexical items and some grammatical structures that may be out of place in today's newswire language. This has the potential of being incompatible with the NLP resources developed for Modern Standard Arabic, which are usually trained on newswire text.

In this paper, we annotate a small corpus of religious Arabic covering three religious domains, with fine-grained morphological segmentation boundaries and segment-based Part of Speech Tagging.

We show that even though the religious corpus is 21 times smaller than the Arabic Treebank sections used in this paper, the segmenter and POS tagger developed using the religious corpus yield much better results than those trained on the ATB. Moreover, a training set that is the concatenation of both the ATB and the religious corpus yields only slightly better results, which shows the need for building a religious Arabic Treebank. Small as it is, the religious corpus we annotate improves segmentation accuracy by 5% absolute (from 90.84% to 95.70%), and POS tagging by 9% absolute (from 82.22% to 91.26) when using gold standard segmentation, and by 9.6% absolute (from 78.62% to 88.22) when using automatic segmentation.

The rest of this paper is divided as follows: Section 2 presents the data we annotated and used in this paper, the methods and the evaluation schemes, section 3 presents the experiments we ran to test the usefulness of the religious corpus, and section 4 concludes and suggests future directions.

2 Data, Methods, and Evaluation

The author of this paper has annotated 3 booklets that cover religious material of enough variety to achieve proper coverage given the small amount of data included. The language variety these texts is written in is more of Classical Arabic than Modern Standard Arabic, and hence the need for the data. The books comprising the data are as follows: (1) ***Al>HAdyv Alnwwyp*** (الأحاديث النووية). This is a book of 50 traditions by Prophet Mohamed selected by Imam Nawawy (1233-1277 AC). The traditions cover a variety of topics with sayings attributed to Prophet Mohamed (571-631 AC). The book will henceforth be referred to as **Nawawy**.

(2) ***mtn >by $jAE*** (متن أبي شجاع) , **Matn** henceforth, is a booklet by the scholar ***>by $jAE*** (-1196 AC) about Islamic law that was intended to be short enough to be memorized by students. The book covers everything from cleanliness to Jihad, and from prayers to adjudication. The book is written in a very concise language.

(3) ***Almnq* mn AlDlAl*** (المنقذ من الضلال), (Eng. The Deliverer from Error), **Munqith** henceforth, is a book by Imam Gazaly (1058-1111) in which he narrates his journey to Sophism. The book focuses on matters of philosophy and belief. It is written in the first person, and addresses virtual listeners.

Table 1 provides basic statistics about the three books.

Book	Words	Types	Segments	seg types
Nawawy	4479	1323	6785	951
Matn	8832	3525	16774	2205
Munqith	14131	4824	23495	2857
Total	27442	**8686**	47054	**4818**

Table 1: basic statistics about the religious corpus

The three books above have been semi-automatically morphologically segmented and pos-tagged by the author of this paper. First, the texts were automatically segmented and tagged then manually checked and corrected. The annotation scheme follows that of the Arabic Treebank (Bies and Mamouri, 2003). The annotation was meant to be as detailed as possible since detailed annotation can be used for deriving many forms of POS tags and word segmentations. The following section details both segmentation annotation and POS annotation.

2.1. Segmentation Annotation

For segmentation annotation, every possible affix, whether inflectional or clitical has been marked as a segment boundary. For example, the word ***fhjrth*** (فهجرته) is annotated as ***f+hjr+t+h***, where ***f*** is a syntactic token, ***hjr*** is a lexical unit, ***t*** is a subject inflection, and the final ***h*** is a pronoun. If the segmentation is ambiguous, then it is done according to the context.

2.2. Part of Speech Annotation

In annotating POS tags, we have opted for a tag set that is as detailed as possible. The tag set works at the segment level and encodes NUMBER, GENDER, DEFINITENESS, MOOD, CASE, and others. For example, the word ***fhjrth*** above is tagged as

f+/CONJ
hjr/NOUN
+***t***+/NSUFF_FEM_SG
+***h***/POSS_PRON_3MS

where ***CONJ*** means conjunction, ***NSUFF_FEM_SEG*** is the Noun Suffix for the Feminine Singular, and ***POSS_PRON_3MS*** is the Possessive Pronoun for the Third Person Masculine Singular. This process is highly context-dependent since the word ***fhjrth*** has at least four other possible POS tag sequences: f/**CONJ**+hjr/**PV**+t/**PVSUFF_SUBJ:3FS**+h/**PVSUFF_DO:3MS**, f/**CONJ**+hjr/**PV**+t/**PVSUFF_SUBJ:1S**+h/**PVSUFF_DO:3MS**, f/**CONJ**+hjr/**PV**+t/**PVSUFF_SUBJ:2MS**+h/**PVSUFF_DO:3MS** and /**CONJ**+hjr/**PV**+t/**PVSUFF_SUBJ:2FS**+h/**PVSUFF_DO:3MS**. This results from the fact that ***hjr*** is both a verb and a noun, ***t*** could be a first person subject pronoun, a second person female subject pronoun, a second person male subject pronoun, or, when affixed to a noun, a singular feminine marker.

2.3. Annotating Assimilated Forms

Arabic has some short (assimilated) forms consisting of a preposition and a pronoun. Table 2 list some of the most common forms in their long and short (naturally occurring) forms.

Our policy of annotating assimilated forms is to go with the conventional written form rather than undo the assimilation. For example, ***Emn*** is annotated as ***E***/PREP+***mn***/REL_PRONOUN instead of ***En***/PREP+***mn***/REL_PRON which is used in the ATB.

Long	Short	English
En mn	Emn	About whom
mn mn	Mmn	From whom
En mA	EmA	About what
mn mA	mmA	From What
ElY y	Ely	On me

Our policy of annotating assimilated forms is to go with the conventional written form rather than undo the assimilation. For example, ***Emn*** is annotated as ***E***/PREP+***mn***/REL_PRONOUN instead of ***En***/PREP+***mn***/REL_PRON which is used in the ATB.

A similar pattern occurs with the definite article ***Al*** when preceded by the preposition ***l***. While the ATB annotates this as ***l***/PREP+***Al***/DET as in the word ***l***/PREP+***Al***/DET+***mjtmE***/NOUN. We do not undo the assimilation and annotate this as ***l***/PREP+***l***/DET+***mjtmE***/NOUN as it occurs in naturally occurring Arabic.

The reason for this is that we do not make use of a morphological analyzer, and once they are segmented and tagged correctly, it's trivial to obtain the original information, although this is hardly needed.

The Arabic Treebank training set has been modified to conform to the same rules of assimilated forms.

3. Experiments

In order to show the usefulness of annotating religious data, we run the following three sets of experiments in which we vary the training set in both segmentation and POS tagging:

1. Train on newswire data and test on the religious data
2. Train on religious data and test
3. Train on a concatenation of the training sets in 1 and 2 above.

We divide the religious data into a training set (80% of the sentences) and a test set (20%). The sentences are assigned randomly to the test and training sets once, and then kept separate. This insures that the test set is the same across all experiments, which allows for proper comparisons between the different experiments.

3.1. Segmentation Experiments

For segmentation, we use the Timbl Memory-based learner (Daelemans *et al.*, 2010) with settings that have been tuned on the ATB data, with a feature representation in which we use the preceding five characters and the following five characters, when present, in a sliding window as features. We use the Timbl IB1 algorithm with similarity computed as overlap, using weights based on gain ratio, and the number of *k* nearest neighbours equal to 1. These settings were reported to achieve an accuracy of 98.15% when trained and tested on standard Arabic Treebank Data (Mohamed, 2010). These experiments also showed that the wider context and part-of-speech tags have only a very limited effect on segmentation quality and that word-internal context alone is enough for producing high quality segmentation.

We run three segmentation experiments:

1. **ATB**: In this experiment, we train on two sections of the ATB (p1v3+p3v2) and test on the religious test set.
2. **Religious**: we train on the Religious 80% and test on the religious 20%
3. **ATB+Religious**: We train on the concatenation of the training sets in 1 and 2, and test on the test set.

For evaluation, we use word level accuracy: a word is correctly segmented if and only if every segment boundary in it is marked correctly. A partially correct segmentation is a wrong segmentation. For example, the word ***fhjrth*** above has to receive the segmentation ***f***+*hjr*+*t*+*h* to be considered correct, and even though ***fhjr+t+h*** has two segments marked correctly, the fact that one segment is wrong renders the whole word wrong.

3.1.1. Segmentation Results and Discussion

Table 3 shows the results of the three segmentation experiments above.

Experiment	Accuracy	Known Word %
ATB	90.84%	55.61
Religious	95.17%	76.70%
ATB+Religious	95.70	80.89%

Table 3: Segmentation Results

With the newswire data as training, the segmentation accuracy is 90.84%. A direct comparison with the Religious-trained segmenter shows a considerable difference of 4.33% in word accuracy. Combining both training sets (ATB+Religious) yields only a slight improvement of 0.53%.

There is a strong indication that the improvement may be attributed to the decrease in the rate of out-of-vocabulary words. While OOV's are 44.80% in the ATB experiment, they drop to 23.3 in the Religious experiment.

3.2. Part of Speech Tagging Experiments

For the POS tagging experiments, we use a memory-based tagger, MBT (Daelemans et al., 1996). The best results were obtained on the ATB data with the Modified Value Difference Metric as a distance metric and with ***k***, the number of nearest instances, = 25. For known words, we use the IGTree algorithm and 2 words to the left, their POS tags, the focus word and its ambitag (list of all possible tags), 1 right context word and its ambitag as features. For unknown words, we use IB1 as algorithm and the unknown word itself, its first 5 and last 3 characters, 1 left context word and its POS tag, and 1 right context word and its ambitag tag as features.

For POS tagging, we use two types of tagging settings:

1. Segmentation-based POS tagging in which the tagging is performed at the segment level. The words are then collected from those segments and the evaluation is performed at the word level. For example, to pos-tag the word ***llmmslmAt***, the word is first segmented into ***l+l+mslm+At***, and each segment is tagged (as in Table 4). Also note that While the segmentation used in the example in Table 4 is gold standard, we do not assume gold standard segmentation and will report results on both gold standard and automatic segmentations.

2. Whole Word Tagging. In this scheme, we do not use any segmentation but rather tag the word as a whole with a composite tag. The word *llmslmAt* thus receives the composite tag *PREP+DET+NOUN+NSUFF_FEM_PL* which has to be produced completely correctly by the tagger for the word to be correctly tagged.

Segment	Gold Tag	Predicted Tag
l	PREP	PREP
l	DET	DET
mslm	NOUN	ADJ
At	NSUFF_FEM_PL	NSUFF_FEM_PL
#	WORD_BOUNDARY	WORD_BOUNDARY
	l/PREP+l/DET+mslm/**NOUN**+At/NSUFF_FEM_PL	l/PREP+l/DET+mslm/**ADJ**+At/NSUFF_FEM_PL

Table 4: Segment-based tagging

The number of segment tags in the ATB training set is 139, while the number of tags in the Religious training set is 117. There are 6 tags in the Religious training set that do not occur in the ATB training set three of which are suffixes of the imperative verb. This shows the more conversational, albeit formal, nature of religious texts.

As far as the test set is concerned, it has 96 segment tags only one of them does not occur in the ATB training set, while 3 tags in the Religious training set do not occur in the test set.

Based on whether the training set comprises the ATB data alone, the religious training alone, or a combination thereof, we have run the following 9 experiments, six of which using segments and the other three with whole words:

1. **ATB GOLD**: Train on the ATB. The test segmentation is gold standard.
2. **ATB AUTO**: Train on the ATB. The test segmentation is automatic.
3. **REL GOLD**: Train on the Religious. The test segmentation is gold standard.
4. **REL AUTO**: Train on the Religious. The test segmentation is automatic.
5. **REL+ATB GOLD**: train on the concatenation of Religious and ATB, test on the gold standard segmentation
6. **REL+ATB AUTO**: train on the concatenation of Religious and ATB, test on the automatic segmentation.
7. **ATBWW:** train on the ATB whole words
8. **RELWW**: train on Religious whole words
9. **RELWW+ATBWW**: the concatenation of the training sets in 7 and 8.

3.2.1. POS Results and Discussion

Table 5(A) shows the results of the POS tagging experiments when tagging on segments, while Table 5(B) shows the results on whole words.

The first thing to notice in the results above is that the ATB-trained tagger performs poorly on religious Arabic. The difference in genre and the high ratio of out of vocabulary words are mainly to blame. While OOV words constitute 44% of the test set when training on the ATB, they are only

23% when training on the religious training set in spite of the fact that the ATB training set is 22 times as big (499884 versus 23001 words).

Experiment	Segment Accuracy	Word Accuracy
ATB GOLD	92.48%	82.82
ATB AUTO		78.62
REL GOLD	95.77%	90.55%
REL AUTO		87.33
REL(*10)+ATB GOLD	96.23	91.26
REL(*10)+ATB AUTO		88.22

Table 5(A): Segment-based POS results

There is also a considerable difference between tagging based on gold standard segmentation and that based on automatic segmentation. This hold true for all experiments, with a difference of 4.2% in the ATB experiment (82.82 vs. 78.62), 3.2% in the REL experiment (90.55 vs. 87.33), and 3% in the REL+ATB experiment (91.26 vs. 88.22). This shows that with more religious data available, the difference could shrink even more.

While segment-based tagging is prone to errors due to the problems resulting from segmentation, another approach is to use whole words with complex tags as units for tagging.

Experiment	Result
ATBWW	78.44%
RELWW	85.90
ATBWW+RELWW	86.96
ATBWW+RELWW*10 (rel train repeated 10 times)	87.24

Table 5(B): Whole word POS results

Results of whole word tagging show more or less the same patterns. The religious-trained tagger outperforms the ATB-trained tagger by 7.5%. The best results are obtained by the concatenation of the religious and ATB training data, repeating the earlier 10 times. This setting achieves an 8.8% absolute improvement over the ATB-trained tagger. This is only about 1% worse than the best-scoring automatic segment-based experiment, and we expect that with more data, the whole word approach would work better than with performing segmentation.

Whole word tagging results are impressive given that the ATB training set has 991 unique tags and the Religious training set has 569. The number of whole word tags in the test set is 324

3.2.2. POS Error Analysis

Due to the many experiments included, it may not be feasible to report on every error in every experiment. We will limit our error analysis to two experiments: ATB GOLD and REL GOLD. We will assume that in the two AUTO experiments, the extra errors are a result of erroneous segmentation.

Table 6 reports on the accuracies of the most common 20 tags in the test set. The top 20 tags count for 90% of all tags with NOUN ranking # 1 at 21.152%, the definite determiner DET # 2 at 11.3%, CONJ # 3 at 9.8%, prepositions PREP # 4 at 9.26 and PUNC # 5 at 8.24%. The worst scoring tags in the ATB experiment are ADJ, PV, NOUN_PROP, REL_PRON and NOUN, while the worst scoring ones in the REL experiment are ADJ, PV, NOUN_PROP, IV, and POSS_PRON_3MS.

Table 7 shows the confusion matrix between the three common low-scoring tags.

Tag	ATB Accuracy	REL Accuracy
NOUN	83.91%	92.08%
DET	99.81%	100%
CONJ	91.40%	100%
PREP	99.50%	98.50%
PUNC	93%	100%
NSUFF_FEM_SG	97.40%	99.35%
PV	71.77%	79.58%
IV	94.18%	88.38%
IV3MS	93%	99.61%
ADJ	66.94%	71.43%
SUB_CONJ	95.19%	99.03%
NEG_PART	100%	100%
PRON_3MS	98.63%	96.58%

POSS_PRON_3MS	94.12%	91.60%
NOUN_PROP	75.12%	82.05%
NUM	91.07%	96.43%
CASE_INDEF_ACC	97.24%	98.17%
REL_PRON	82.02%	94.38%] \
PRON_3FS	98.36%	98.36%
NSUFF_FEM_PL	100%	100%

Table 6: Frequent tag accuracies

Tag	ATB Confusions	REL Confusions
ADJ	NOUN 21.63% NOUN_PROP 11.48%	NOUN 25.31 NUM 1.63 NOUN_PROP 0.41
PV	NOUN 18% NOUN_PROP 7.8% PREP 1.2%	NOUN 12.91 IV 2.7 ADJ 1.8
NOUN_PROP	NOUN 19.66% ADJ 4.27%	NOUN 14.52% IV 0.85%

Table 7: Most common POS confusions

4. Related Work

To our knowledge, there exists no work that handles the morphological segmentation and part of speech tagging of religious Arabic, but some works are related which focused mostly on the Quran. Alhadj (2009) built a POS tagger for traditional Arabic with the ultimate aim of using the tagger for building a Quranic linguistic database. He trained his tagger on "Albayan-wa-tabyin" , a book by Al-Jahiz. However, the book is a literary one focusing on rhetoric, and the POS tagset used was very limited (13 tags). There is no clear evaluation of Quranic Arabic in the paper.

Another effort, also targeting the Quran, is that of the Quranic Arabic Corpus (corpus.quran.com) (Dukes and Buckwalter: 2010). The QAC is a comprehensive database of the Quran including morphological analysis, part of speech tagging, and dependency parsing. The Quranic Arabic Corpus differs from the work in this paper in that it is limited to the Quran, while we try to leverage a corpus and tools for many varieties of religious Arabic as attested by the selection of the three books in our tiny corpus. The POS tagset we use is generally more detailed than the one used in the QAC since we also segment and tag inflectional affixes, although their treatment of particles seems to be more appropriate, and we will try to include it in our future work.

Arabic POS tagging has long been an important topic in Arabic NLP in general, and several approaches exist. Habash and Rambow (2005) perform full morphological analysis that produces segmentation and POS tags as by-products. Mohamed and Kuebler (2010a, 2010b) and Kuebler and Mohamed (2011) treat segmentation as per letter classification task and perform POS tagging at the segment level where inflectional as well as syntactically functional tags are segmented and tagged. Diab et al (2007) and Diab (2009) use a pieplined approach in which they first perform tokenization then POS tagging using support vector machines without the use of a morphological analyzer. Kulick (2010) avoids the pieplined approach by performing simultaneous tokenization and POS tagging with a small tag set and reports promising results.

3 Conclusion

We have presented a small corpus of religious Arabic, and the results of word segmentation and POS tagging. We have compared the results obtained by training a segmenter and a POS tagger, and shown that even though the religious corpus is tiny, it produces better results than the ATB-trained segmenter and tagger. It is worth noting that even if we obtain a much larger newswire corpus for training, the results may not be better. We have checked the coverage in a 148,363,649 word portion of the Arabic Gigaword corpus (Graff et al, 2006) , and found that the OOV rate is 22.82% at the word type level and 9.32% at the token level.

Religious Arabic thus requires its own Treebank. We will work on adding more data to the current "tiny" selection making sure to cover the various aspects of religious Arabic as well as add more layers of annotation.

References

Mona Diab. 2009. Second generation tools (AMIRA 2.0): Fast and robust tokenization, pos tagging, and base phrase chunking. In Proceedings of 2nd International Conference on Arabic Language Resources and Tools (MEDAR), Cairo, Egypt, April.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2007. Automatic processing of Modern Standard Arabic text. In Abdelhadi Soudi, Antal van den Bosch, and Gunter Neumann, editors, Arabic Computational Morphology, pages 159–179. Springer.

Kais Dukes and Timothy Buckwalter. 2010. A Dependency Treebank of the Quran using Traditional Arabic Grammar. In Proceedings of the 7th International Conference on Informatics and Systems (INFOS). Cairo, Egypt.

David Graff, Ke Chen, Junbo Kong, and Kazuaki Maeda (2006). The Arabic Gigaword Corpus. Second Edition. LDC Catalog No. LDC2006T02

Yahya O. Mohamed Elhadj. 2009. Statistical Part-of-Speech Tagger for Traditional Arabic Texts. Journal of Computer Science 5 (11): 794-800, 2009 . Science Publications.

Nizar Habash, Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 573-680.

Sandra Kuebler and Emad Mohamed. 2011. Part of speech tagging for Arabic. Natural Language Engineering.

Seth Kulick. 2010. Simultaneous tokenization and part-of-speech tagging for Arabic without a morphological analyzer. Proceeding of ACLShort '10 Proceedings of the ACL 2010 Conference Short Papers. Pages 342-347.

Emad Mohamed and Sandra Kübler. 2010a. Arabic part of speech tagging, Proceedings of LREC 2010, Valetta, Malta.

Emad Mohamed and Sandra Kübler. 2010b. Is Arabic part of speech tagging feasible without word segmentation? Proceedings of HLT-NAACL 2010, Los Angeles, CA.

Exploiting Wikipedia as a Knowledge Base for the Extraction of Linguistic Resources: Application on Arabic-French Comparable Corpora and Bilingual Lexicons

Rahma Sellami
ANLP Research Group
Laboratoire MIRACL
University of Sfax, Tunisia
rahma.sellami@gmail.com

Fatiha Sadat
UQAM, 201 av. President Kennedy,
Montreal, QC, H3X 2Y3, Canada
sadat.fatiha@uqam.ca

Lamia Hadrich Belguith
ANLP Research Group
MIRACL Laboratory
University of Sfax, Tunisia
l.belguith@fsegs.rnu.tn

Abstract

We present simple and effective methods for extracting comparable corpora and bilingual lexicons from Wikipedia. We shall exploit the large scale and the structure of Wikipedia articles to extract two resources that will be very useful for natural language applications.

We build a comparable corpus from Wikipedia using categories as topic restrictions and we extract bilingual lexicons from inter-language links aligned with statistical method or a combined statistical and linguistic method.

1 Introduction

Multilingual linguistic resources are usually constructed from parallel corpora. Unfortunately, parallel texts are scarce resources: limited in size, language coverage, and language register. There are relatively few language pairs for which parallel corpora of reasonable sizes are available.

The lack of these corpora has prompted researchers to exploit other multilingual resources such as comparable corpora. Comparable corpora are "sets of texts in different languages that are not translations of each other" (Bowker and Pearson, 2002), but contains texts from the same domain.

Comparable corpora have several obvious advantages over parallel corpora. They are available on the Web in large quantities for many languages and domains and many texts with similar content are produced every day (e.g. multilingual news feeds) (Skadiņa et al, 2010), but they are not organized.

Also, bilingual lexicons are the key component of all cross-lingual NLP applications such as machine translation (Och and Ney, 2003) and cross-language information retrieval (Grefenstette, 1998).

Parallel texts – as the most important resource in statistical machine translation (SMT) – appear to be limited in quantity, genre and language coverage. Providing more comparable corpora essentially boosts the coverage and the quality of machine translation system, especially for less-covered languages and domains.

In this paper we describe the extraction process of large comparable corpora and bilingual lexicons for Arabic and French language from a multilingual web-based encyclopedia, Wikipedia.

We propose to build bilingual resources as follows: first comparable corpora from Wikipedia using categories and languages as restrictions; next two bilingual lexicons extracted from titles of articles that are related by inter-language links and aligned by a statistical based method and a combined statistical and linguistic-based method.

The best extracted lexicon will be used to improve the mining of different levels of parallelism from our comparable corpora.

The content of this paper is summarized as follows: Section 2 describes some characteristics of Wikipedia that makes it a source of multilingual resources extraction. Section 3 presents a brief overview of previous works on comparable corpora and bilingual lexicon extraction from Wikipedia. In sections 4 and 5, we present and evaluate our work of mining Arabic-French comparable corpora and bilingual lexicon from Wikipedia. We conclude the present paper in section 6.

2 Characteristics of Wikipedia

In the following sub-section, we shall describe some of the interesting characteristics of Wikipedia that make the encyclopedia an invaluable resource for knowledge mining.

Wikipedia is an online encyclopedia under the non-profit Wikimedia Foundation. Unlike ordinary encyclopedias, the Wikipedia project is based on the wiki concept (Leuf and Cunningham, 2001), thus anyone can contribute by creating, editing or improving the articles.

2.1 Wikipedia Coverage

Wikipedia currently (2012) contains more than 22 million articles among which 1 259 482 are written in French and 179 291 are written in Arabic language[1]. These articles cover different categories such as arts, geography, history, society, science and technology. Wikipedia articles cover many domain-specific concepts as well as named entities (i.e. proper nouns such as names of persons), including even latest topics since Wikipedia is being updated all the time.

2.2 Wikipedia Link Structure

- Inter-language Links

An inter-language link in Wikipedia is a link between two articles in different languages. An article has usually one inter-language link for each language.

Inter-language links are created using the syntax *[[language code:article title]]*. The *language code* identifies the language in which the target article is written and *Article title* is the title of the target page (e.g. [[fr:Lac Tchad]]). Since the titles of all Wikipedia articles in one language are unique, that information is sufficient to identify the target page unambiguously.

- Redirect Pages

A redirect is a page, which has no content itself, but sends the reader to another article, section of an article or page, usually from an alternative title. A redirect page can be created by writing the text *#REDIRECT [[article title]]* at the top of the article where *article title* denotes the name of the target page.

Redirect pages are used in particular for Adjectives/Adverbs point to noun forms (e.g. *Treasonous* redirects to *Treason*), Abbreviations (e.g., *DSM-IV* redirects to *Diagnostic and Statistical Manual of Mental Disorders*), Alternative spellings or punctuation (e.g. *Al-Jazeera* redirects to *Al Jazeera*), etc.

- Link Texts

This is a link to another page in Wikipedia. The link text can correspond to the title of the target article (the syntax will be: *[[article title]]*), or differ from the title of the target article (with the following syntax: *[[article title | link text]]*).

As a rich and free resource, Wikipedia has been successfully used as an external resource in many natural language processing tasks (Buscaldi and Rosso, 2006; Mihalcea, 2007; Nakayama et al., 2007).

3 State of the Art

In accordance with fast growth of Wikipedia, many works have been published in the last years focused on its use and exploitation for multilingual tasks in natural language processing: in this paper, our main concern is the use of Wikipedia as a source of comparable corpora and bilingual lexicon extraction.

Li et al. (2010) consider Wikipedia as a comparable corpus, they align articles pairs based on inter-language links for the extraction of parallel sentences. Patry and Langlais (2011) also concentrate on documents pairs that are linked across language for extracting parallel documents. However, Smith et al. (2010) and Mohammadi and QasemAghaee (2010) use inter-language link to

[1] http://meta.wikimedia.org/wiki/List_of_Wikipedias

identify aligned comparable Wikipedia documents. Sadat (2010) proposes an approach to build comparable corpora from Wikipedia encyclopedia. First, the author considers a preliminary query Q in a source language to input in Wikipedia search engine. The resulting document is used as a first document for the corpus in the source language. The usage of the inter-language link in the target language for this document leads to a corpus in a target language. Following this first step and exploiting the links in the same document as well as the inter-language links, comparable corpora are built for the query Q.

Otero and Lopez (2010) propose an automatic method to build comparable corpora (CorpusPedia) from Wikipedia using Categories as topic restrictions. Given two languages and a particular topic, their strategy builds a corpus with texts in the two selected languages, whose content is focused on the selected topic. Again, Otero and Lopez (2011) propose two strategies to build comparable corpora from Wikipedia: The first one (non-aligned corpus) extracts those articles in two languages having in common the same topic. It results in a non-aligned comparable corpus, consisting of texts in two languages. The second strategy (aligned corpus) extracts pairs of bilingual articles related by inter-language links, with the condition that at least one of both contains a required category. It results in a comparable corpus with aligned articles. The input of the two strategies is CorpusPedia developed by Otero and Lopez (2010).

Plamada and Volk (2012) demonstrate the difficulty to use Wikipedia categories for the extraction of domain-specific articles from Wikipedia. They propose an Information Retrial (IR) approach in order to achieve a solution to this task and they identify articles that belong to the Alpine domain based on this approach.

Skadina et al., (2012) developed a technique to find comparable Wikipedia texts based on inter-language link. First, they extract all document pairs connected by inter-language link and share the same topic. Then, they filter out non-comparable articles; they measure the similarity of document pairs by performing cross-lingual sentences alignment.

Several works have a common characteristic: their comparable corpora are composed from articles related by inter-language links that may share or not the same topic. However, our work is based on the definition of comparable corpora, a set of texts that share some criteria without being in mutual translation. We constructed a comparable corpus from articles that share at least one topic, but are not necessary related by any inter-language link.

Other works on the extraction of bilingual lexicons from Wikipedia are described as follows: Adafre and Rijke (2006) created a bilingual dictionary (English-Duch) from Wikipedia in order to help construct a parallel corpus. The authors demonstrated that the bilingual lexicon approach for constructing a parallel corpus is more accurate and efficient than the machine translation based approach. Bouma et al. (2006) extracted bilingual terminology for creating a multilingual question answering system (French-Dutch). In addition, Decklerck et al. (2006) used bilingual terminology for translating ontology labels; they used only inter-language links for bilingual terminology extraction.

What all researches have in common is the fact that they use only inter-language links for extracting bilingual terminology. However, Erdmann et al. (2008) analyze not only the inter-language link of Wikipedia, but also exploit redirects links and link texts to build an English-Japanese dictionary. The authors have shown the contribution of using Wikipedia compared to parallel corpus for the extraction of a bilingual dictionary. This contribution appears especially at the wide coverage of terms.

Sadat and Terrasa (2010) propose an approach for extracting bilingual terminology from Wikipedia. This approach, first, extract pairs of words and translations from different types of information, links and text of Wikipedia, then, use linguistic information to reorder the relevant terms and their translations. More recently, Ivanova (2012) evaluates a bilingual bidirectional English-Russian dictionary created from titles of Wikipedia articles. She explored the inter-language links and redirect pages methods described in (Erdmann et al., 2008) in order to create English-Russian Wiki-dictionary. The author demonstrates that Machine translation experiments with the Wiki-dictionary incorporated

into the training set resulted in the rather small, but statistically significant drop of the quality of translation compared to the experiment without the Wiki-dictionary. However, using the test set collected from Wikipedia articles, the model with incorporated dictionary performed better.

4 Comparable Corpora Extraction

4.1 Extraction Process

In this paragraph, we describe our method for building a comparable corpus from Wikipedia articles. This method extracts those articles in two languages having in common the same topic where the topic is represented by a category and its translation.

The process to extract Arabic-French comparable corpora from Wikipedia is described as follows:

First step consists on downloading French and Arabic Wikipedia database (January / February 2012) from http://download.wikimedia.org.

Second, all Arabic topics that have French translations are extracted from Wikipedia articles. An example is the phrase title in Arabic "تصنيف: لاعبو كرة مضرب ألمان", that leads to a French translation *"joueur allemand de tennis"*, when following the syntax of inter-language links.

Third, for each pair of topics, we extract all Arabic and French articles which have a link text to the selected topic.

Fourth step consists on cleaning the extracted articles by removing Wikipedia markups.

Through these steps, we get a comparable corpus of texts in Arabic and French languages sharing the same topic. The comparable corpus covers all topics that exist in Wikipedia.

We should note that there are articles in Arabic, respectively in French, with no corresponding version in French, respectively Arabic.

4.2 Experiments and Results

We download Arabic and French Wikipedia database (January/February 2012) in XML format from http://download.wikimedia.org.

We extract 20 533 Arabic topics that have translation in French language.

In order to have an idea about the size of our corpus, we present the number of Arabic and French articles for the first ten extracted topics. Table 1 summarizes the quantitative description of generated corpora.

Category	Number of Arabic articles	Number of French articles
بحيرات / Lac 'Lake'	41	9
حروب / Guerre 'War'	51	66
رؤساء مصر / Président d'Égypte 'President of Egypt'	5	5
نازية / Nazisme 'Nazism'	11	52
فلك / Astronomie 'astronomy'	255	47
فلاسفة / Philosophe 'philosopher'	40	5
عناصر كيميائية / Élément chimique'chemical element'	168	165
لغات برمجة / Langage de programmation 'Programming language'	39	260
قارات / Continent 'Continent'	29	12
لغات / Langue 'language'	80	32
Total	**719**	**653**

Table 1. Number of Arabic and French articles for the first ten extracted topics.

The table shows that there are significant differences in term of the size among the Arabic and French language, e.g. 41 Arabic articles are sharing the category "بحيرات/Lac 'Lake'" against only 9 in French. However the difference between Arabic and French is less expected since we extract these topics from an Arabic database to seek French articles that share the same topics.

5 Bilingual Lexicon Extraction

5.1 Extraction Process

We propose to use a simple but effective method for bilingual lexicon extraction; it exploits inter-language links between Wikipedia articles to extract Arabic terms (simple or multi-word) and their translations into French. We then use a statistical approach for aligning words of compound terms. Also, linguistic-based filtering based on the part of speech can be applied in order to keep pertinent translation candidates.

We analyze all inter-language links in Wikipedia to create an Arabic-French lexicon. These links are created by the authors of the articles; we assume that the authors correctly positioned these links. Also, an article in the source language is linked to a single article in the target language. Therefore, possible problems of ambiguity in the extraction of pairs of titles are minimized.

We start by downloading Wikipedia database (January 2012) in XML format and thus extracting about 104 104 (Arabic-French) inter-language links. Each link corresponds to a pair of Arabic-French titles.

Some titles are composed of a simple word, while others are composed of multi words. We performed an alignment step in order to have a lexicon consisting only of simple words.

Before aligning these titles, we proceed to preprocessing them. The preprocessing step consists of removing all Arabic and French stop words.

The step of word alignment presents several challenges. First, the alignments are not necessarily contiguous. Two consecutive words in the source sentence can be aligned with two words arbitrarily distant from the target sentence. This is called distortion. Second, a source language word can be aligned to many words in the target language; that is defined as fertility.

The alignment of words of each title is based on IBM models [1-5] (Brown et al., 1993) in combination with the Hidden Markov Model (Vogel et al., 1996). These standard models have already proven their effectiveness in many researches.

The five IBM models estimate the probability P(fr|ar) and P(ar|fr), for which fr is a French word and ar is an Arabic word. Each model is based on the parameters estimated by the previous model and incorporates new features such as distortion, fertility, etc.

The Hidden Markov Model (HMM usually appointed) (Vogel et al., 1996) is an improvement of IBM2 model. It explicitly models the distance between the alignment of the current word and an alignment of the previous word.

We used the open source toolkit GIZA++ (Och and Ney, 2003) that is an implementation of the original IBM models. Thus, GIZA++ was run in both directions to obtain two GIZA++ alignments (Arabic to French and French to Arabic).

Next, two different methods have been exploited in order to filter the two lexicons. First, a combined statistical method with a grow-diag-final heuristics will keep the intersection of the two alignments and thus add additional alignment points. In total we extracted 224 379 translation pairs.

Second a linguistic-based method will keep translation candidates with corresponding part of speech tags of both Arabic and French alignment and will discard non pertinent translations. Stanford Part-Of-Speech Tagger[2] is used for both languages. In total we extracted 235 938 translation pairs.

5.2 Evaluation

Since the titles of Wikipedia articles are usually nouns, our lexicon does not contain verbs.

We calculated the standard criteria precision to measure the accuracy of our methods for the extraction of Arabic-French lexicon from Wikipedia. Our result is based on the precision measure that calculates how many of the extracted translation candidates are correct, as follows:

$$\text{precision} = \frac{|\text{Extracted correct translations}|}{|\text{All extracted translation candidates}|}$$

It is not trivial to estimate the total number of correct translations for a term. Since it cannot be calculated automatically, we conduct a manual

[2] http://nlp.stanford.edu/software/tagger.shtml

evaluation with a support from an expert. We calculate the precisions of our two lexicons based on the candidate translations of 50 words and we compare it to the precision of the online LAROUSSE[3] dictionary.

Table 2 summarizes a comparative description of generated lexicons.
The combined statistical and linguistics based methods that is enhanced with part of speech (POS) filtering achieved a better precision than the stand-alone statistical method.

Statistical method + POS	
candidates	precision
189	80.15%
Statistical method	
candidates	precision
237	76.02%
LAROUSSE	
candidates	precision
66	95,45%

Table 2. Evaluation based on the candidates translations of 50 French words.

The coverage value represent the number of candidates translations, it is 224 379 for the lexicon based on the statistical method and 235 938 for the lexicon based on a combined statistical and linguistics based method using the part of speech filtering.

Mistranslations of our Arabic-French lexicons are mainly due to the fact that some articles' titles are introduced in language other than Arabic (e.g. cv / cv), mostly in English and some translations candidates are transliteration of Arabic word (e.g. Intifada / انتفاضة). Also, we detected alignment errors (e.g. نفسية/ diagnostic). Other errors are due to the fact that pairs of titles are not accurate translations but refer mainly to the same concept (e.g. Christmas / عيد).

6 Conclusion and Future Work

The semi-structured information underlying Wikipedia turns out to be very useful to build multilingual resources such as comparable corpora, parallel corpora, multilingual lexicons and ontologies.

In this paper, we presented our preliminary work on mining Wikipedia for the extraction of comparable corpora and bilingual lexicons. Our major goal is to exploit the multilingual aspect of Wikipedia for Statistical Machine Translation.

On the one hand, we exploit the classification of articles with categories corresponding to topics or genders to extract an Arabic-French comparable corpus. On the other hand, we exploit the network of inter-language links to create an Arabic-French bilingual lexicon.

Unlike previous works that exploit inter-language link to construct comparable corpora, we have tried to build our comparable corpus by selecting Arabic and French articles that share at least one topic. This strategy improves the coverage of comparable corpora. Indeed, even articles that share the same topic despite not related by any inter-language link may contain parallel sentences.

Also, the proposed methods of bilingual lexicon extraction are effective despite its simplicity. We extract Arabic and French articles' titles based on inter-language links between Wikipedia articles. We align words of these titles based on statistical method first; then based on a combined statistical and linguistics based method using the part of speech filtering.

We have reached encouraging levels of precision and coverage, mainly for the second method. These levels exceed respectively 90% and 235 938 pairs of translations for the combined statistical and linguistics-based method using the part of speech filtering.

Finally, in future work, we will define an evaluation protocol to measure the degree of comparability between texts of our comparable corpus. For this purpose, we will make use of techniques described in (Otero and Lopez, 2007) which take advantage of the translation equivalents inserted in Wikipedia by means of inter-language links. We also plan to expand the coverage of our lexicon by exploiting other links like Wikipedia redirect pages and link text. We also envisage using the lexicon to extract Arabic-French parallel corpus from our comparable corpus. The parallel

[3] http://www.larousse.fr/dictionnaires/francais-arabe/

corpus will be used as training data for Statistical Machine Translation.

References

Adafre, S. F. and De Rijke, M. 2006. Finding Similar Sentences across Multiple Languages in Wikipedia. In Proceedings of the EACL Workshop on NEW TEXT Wikis and blogs and other dynamic text sources, pages 62–69.

Alexandre Patry and Philippe Langlais. 2011. PARADOCS : Identifying Parallel Documents from a Large Bilingual Collection of Texts: Application to Parallel Article Extraction inWikipedia. Proceedings of the 4th Workshop on Building and Using Comparable Corpora, 9th Annual Meeting of the Association for Computational Linguistics, Portland, 2011.

Bouma, G., Fahmi, I., Mur, J., G. Van Noord, Van Der, L., and Tiedemann, J. 2006. Using Syntactic Knowledge for QA. In Working Notes for the Cross Language Evaluation Forum Workshop.

Bowker Lynne and Pearson Jennifer. 2002. Working with Specialized Language: A Practical Guide to Using Corpora. Routledge, London/New York.

Brown Peter, F., Pietra, V. J., Pietra, S. A., and Mercer, R. L. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. IBM T.J. Watson Research Center, pages 264-311.

Buscaldi D.and Rosso P.. 2006. Mining Knowledge from Wikipedia for the Question Answering Task. Proceedings of the 5th International Conference on Language Resources and Evaluation.

Declerck, T., Perez, A. G., Vela, O., Z., and Manzano-Macho, D. 2006. Multilingual Lexical Semantic Resources for Ontology Translation. In Proceedings of International Conference on Language Ressources and Evaluation (LREC), pages 1492 – 1495.

Erdmann, M., Nakayama, K., Hara, T. Et Nishio, S. 2008. A bilingual dictionary extracted from the wikipedia link structure. In Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA) Demonstration Track, pages 380-392.

Grefenstette, G. 1998. The Problem of Cross-language Information Retrieval. Crosslanguage Information Retrieval. Kluwer Academic Publishers.

Inguna Skadiņa A , Ahmet Aker B , Voula Giouli C , Dan Tufis D , Gaizauskas B , Madara Mieriņa A , Nikos Mastropavlos C. 2010. A Collection of Comparable Corpora for Under-resourced Languages. Fourth International Conference HUMAN LANGUAGE TECHNOLOGIES . “Athena”, Greece.

Isaac Gonzalez Lopez and Pablo Gamallo Otero. 2010. Wikipedia as multilingual source of comparable corpora. In Proceedings of the LREC 2010, Malta.

Ivanova Angelina , 2012. Evaluation of a Bilingual Dictionary Extracted from Wikipedia. InProceedings, 5th Workshop on Building and Using Comparable Corpora (BUCC), Istanbul, Turkey.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In Proceedings of the Human Language Technologies/North American Association for Computational Linguistics, pages 403–411.

Leuf B. and Cunningham W. 2001. The Wiki Way: Collaboration and Sharing on the Internet. Addison-Wesley.

Magdalena Plamada and Martin Volk. 2012. Towards a Wikipedia-extracted Alpine Corpus. 5th Workshop on Building and Using Comparable Corpora at LREC 2012. Istanbul.

Mehdi Mohammadi and Nacer QasemAghaee, 2010. Building Bilingual parallel Corpora based on Wikipedia. In Proceedings of the 2010 Second International Conference on Computer Engineering and Applications - Volume 02, ser. ICCEA ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 264–268.

Mihalcea R. 2007. Using Wikipedia for Automatic Word Sense Disambiguation. Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL-HLT 2007).

Min-Hsiang Li, Vitaly Klyuev and Shih-Hung Wu. 2010. Multilingual sentence alignment from Wikipedia as multilingual comparable corpora . Proceedings of the 13th International Conference on Humans and Computers. Japan.

Mohammadi M. and QasemAghaee N.. 2010. Building bilingual parallel corpora based on Wikipedia. International Conference on Computer Engineering and Applications, 2:264–268.

Nakayama K., Hara T.and Nishio S. 2007. Wikipedia Mining for An Association Web Thesaurus Construction. Proceedings of the 8th International Conference on Web Information Systems Engineering.

Och, F.J. and Ney, H. 2003. A systematic comparison of various statistical alignment models. Computational Linguistics, pages 19–51, March.

Pablo Gamallo Otero.and Isaac Gonzalez Lopez. 2011. Measuring comparability of multilingual corpora extracted from Wikipedia , Proceedings of Workshop on Iberian Cross-Language NLP tasks (ICL-2011) , September 2011, Huelva, Spain.

Sadat, F. et Terrassa, A. 2010. Exploitation de Wikipédia pour l'Enrichissement et la Construction des Ressources Linguistiques. TALN 2010, Montréal.

Skadiņa, I., Aker, A., Glaros, N., Su, F., Tufiş, D., Verlic, M., Vasiļjevs, A. and Babych, B. 2012. Collecting and Using Comparable Corpora for Statistical Machine Translation, in Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey.

Vogel, S., Ney H. and C. Tillmann .1996. HMM-based word alignment in statistical translation. In Preceding of the Conference on Computational Linguistics, pages 836–841, Morristown, NJ, USA.

Author Index