TOOLS FOR ARABIC PEOPLE NAMES PROCESSING AND RETRIEVAL
A Statistical Approach

Ali Y. Salhi, Adnan H. Yahya
Faculty of Information Technology, Birzeit Universiy, Birzeit, Palestine
asalhi@birzeit.edu, yahya@birzeit.edu

Keywords: Arabic People Names, Statistical Databases, Names Correction, Names Translation, Names Gender Detection, Names Extraction, Natural Language Processing.

Abstract: Arabic web content has been rapidly growing, generating a need for tools to overcome the many challenges of processing and retrieving Arabic content: challenges related to Arabic Language Processing, Search and Query Analysis. An important part of dealing with Arabic digital content is processing and analyzing Arabic people names. This paper reports on our work aimed at designing name pre-processing tools that are able to efficiently identify and process Arabic people names in queries and documents. We try to address challenges such as Name Gender Detection, Translation (Arabic to English), Correction, Auto Suggestion and Extraction from text. All through, we employ a statistical approach based on data obtained from High School student names lists in Palestine and Birzeit University student names lists. Based on this information we constructed different types of databases of Arabic names and used them as the infrastructure for the well structured names tools which are capable of being integrated into existing web search engines and document processing systems. We have been experimenting with some of the developed tools in our online application process at Birzeit University, with encouraging preliminary results.

1 INTRODUCTION

In the web, people names are used in user profiles, registrations, articles, forms... etc. One of the main problems facing writing a certain Arabic name is spelling. People with same name write it differently, be it due to ignorance or habit. Take the name ديمة for example, some write it ديما while others write it ديمه . Also translating the name might lead to Dima, Deema, Demah or Dimah, with no clear certainty of which is correct and which is wrong. In some cases it's a matter of common misspelling as in the case of أحمد, when people mistakenly write it احمد by removing the "Hamza" from the first letter. The spelling problem mainly arises in applications such as passports filling systems where you can find different spellings for the same name. It is even possible to find the same name of an individual appearing in different spellings in different user profiles (passport, Tawjihi, and university records).

When the process of filling forms was manual it was hard to follow how people write their names or give them suggestions as to how to write it properly, say in an attempt to unify names writing. Currently the use of the web and online forms (whether it's on the Internet or Intranet networks or on private devices) offers a better chance for fixing, enhancing and introducing names processing tools that help unify names writing by giving spelling options, possible common translations and autosuggestion while writing.

Names are also used in search. When searching for أحمد شوقي one expects to get results with أحمد شوقي in all its (miss)spellings. Also the search engine should be able to detect English articles with the name mentioned in its English form, in this case Ahmad might also be Ahmed and Shawqi might also be Shawqy or Shawqee and so on.

The main focus of this paper is to report on our work aimed at designing Arabic people names processing tools that may help simplify and enrich the Arabic search experience on one hand and the filling of e-forms on the other. We believe that some of the challenges Arabic people names pose for natural language processing and information retrieval include: names gender detection, translation (Arabic to English), correction, autosuggestion and names extraction. We will address each of these issues employing a statistical/database-based approach relying on data obtained from variety names sources. Based on corpus statistics we constructed databases of Arabic names and their frequencies, different types of tables were built and used as an infrastructure for the well structured people names tools which are capable of being integrated into existing web search engines and document processing systems. Such tools will help enhancing connectivity between Arabic articles by defining key people names shared among different articles for example or by applying translation tools to connect different Arabic/English articles together.

Even though our work is focusing on a specific part of text (people names) some of the tools described in this paper can be applied (with proper modifications) to more general categories of text. Also we recognize the regional accent of our infrastructure that may bias processing in favour of the local dialect. Basing the infrastructure on other regional data or consolidating data from different regions may create more robust systems. We even noticed some regional and

generational biases in the Palestinian context. But that is not an issue that we can elaborate on here.

2 BACKGROUND

2.1 Arabic Content and Awareness

There is a global awareness among young people about the importance of Arabic content sharing and creating an online identity resulting from increasing blogging and social networks use, plus forums activities that use Arabic as its main language. On October 23, 2010, ictQATAR and Creative Commons hosted "Digitally Open: Innovation and Open Access Forum". The forum addresses how innovation can push and increase sharing and openness of Arabic content[1]. Earlier, Mohammed bin Rashid Al Maktoum Foundation in Dubai started the Sawaed program focusing on developing the capabilities of talented Arab entrepreneurs by providing non-refundable grants. A major recent focus of Sawaed program is the development of online Arabic content[2]. Also, in 2007 the Information and Communication Technology Division (ICTD) at UN-ESCWA launched a project on "Promotion of the Digital Arabic Content Industry through Incubation"[3]. This awareness and calls for Arabic content increase are just some examples. Development of online Arabic content increased the awareness about the lack in tools and content management services needed for indexing, analyzing and processing the Arabic content. Without such tools, the increased awareness may direct the attention of users to other languages and their Arabic based activities may be fragmented due lack of connectivity and possibilities of integration. Access to multilingual data may get more complicated.

2.2 Tools for Improved Search and Arabic Content Management

Search Engines depend on query processing systems to parse and interpret user input. Search Engines need to understand the user intention so as to provide satisfactory results sought by that user. Query processing tools such as Query Suggestion, Cross Language Query Suggestion (Suggestion and translating to different languages), Web and Query Categorization, Query Analysis and others help improve the search experience and help manage Arabic content. Arabic names tools can contribute in increasing the correctness of data in e-forms filling and enhancing the user experience when performing searches. This is done by providing language support tools such as name spell checking, auto suggestion and query expansion, names correction and others. In the following sections we will discuss some of our work on developing Query Analysis/E-form filling tools focusing on people names.

3 NAMES TOOLS RESOURCES

In order to build statistical based tools for Arabic people names in documents and queries we need to define and prepare names databases to serve as the base for the tools we are building. First we need to build a database that holds names with a statistical indication about the use of each name in the underlying data (frequency of appearance and use). Database rows should have a predefined gender/classification (Male, Female, and Family). Another database maps between Arabic names and their possible translations into English with a statistical description of each English translated form.

The raw data used to build these databases is obtained from Palestinian General High School Certificate Exam (Tawjihi) student lists for the years of 2005, 2007, 2008, 2009 and 2010 (Currently our data reflects West Bank names only) and Birzeit University students and employees names (from 2003 till 2010).

Palestinian Tawjihi list is obtained from the Palestinian ministry of education, which releases the results of the Tawjihi exam in *"xls"* (Microsoft excel) format with student names (first, second, third, last/family name), city, school and exam score. What concern us here are the names fields.

Birzeit list is obtained from Birzeit University, which has the names of all students and employees from 2003 till 2010. The list contains a bag of student name tuples derived from the full students names at the university without order for privacy reasons, and a tuple may be a first name, father name, grandfather name or a family name. Each tuple (repetition allowed) holds a translation to English as well as the gender (Male, Female or Family); where first names can be male/female and last names are family names (some may occur as male/female in other tuples as well). The data corresponds to real records so names may hold different translations and gender values. The list has around 117,000 records with translations and genders. Check Table 1 for a sample of Birzeit list.

From Tawjihi and Birzeit lists the following tables were obtained:

3.1 Male Names Table

This is a table that holds all male names only, and to build this table we filtered all male names from Birzeit list by selecting all names with gender equal to *male* then we added male names extracted from Tawjihi list.

The process of extracting male names from Tawjihi list is not as easy as in the case of Birzeit list. In Tawjihi list first we need to parse 2nd and 3rd fields in students names, those fields are considered male names by default since they reflect father and grandfather names only. After parsing these fields we used the obtained male names to filter male names from the students' 1st names (1st name field holds names that can be a male or a female). This process filters out male names that appear in 2nd and 3rd fields and repeated

in the 1st field, leaving us with a list of names which are either females or males and didn't appear as father\grandfather names field, something unlikely to happen.

Table 1: Birzeit University List Preview

#	Arabic Name	English Name	Gender	#	Arabic Name	English Name	Gender
1	فايزة	Fayze	Female	11	هند	Hind	Female
2	ريم	Reem	Female	12	أسامة	Osama	Male
3	محمود	Mahmoud	Male	13	نداء	Nida2	Female
4	رنا	Rana	Female	14	حسين	Hussein	Male
5	سامي	Sami	Male	15	أمل	Amal	Female
6	نائلة	Naela	Female	16	قصي	Qussay	Male
7	إيمان	Eman	Female	17	بشار	Bashar	Male
8	قدر	Qadar	Family	18	جميل	Jamil	Male
9	هبة	Hiba	Female	19	رشا	Rasha	Female
10	سجى	Saja	Female	20	علي	Ali	Male

However there might be names such as ضياء، جهاد، نور، and so on that might be female or male names. To try to give a fair judgment about such names we assumed that any multi classification (gender) name is considered to be with a male classification if appeared in 2nd or 3rd name field in Tawjihi list (regardless the appearance in 1st field), if the name appeared in 1st field (a first name) only then it's considered to be a female name. Now how multi classification names are detected? By assuming that any 2nd and 3rd field names that are considered to be female in Birzeit list (since Birzeit list has predefined classification) and is found as a 1st name in Tawijhi list is a multi classification name.

Then male names are processed to only save unique names with a frequency counter that gives the number of occurrences of each name. The number of distinct male names obtained is 3570 (without repetition of same form) from a total of 365445 names. However the value doesn't really reflect the real number of different male names, the table needs some filtration to remove compound names such as "أحمد عيسى" or "محمد حسين" and converting "عبدالله" to "عبد الله" (with a space in the middle) which are used as compound names for individuals plus a filtration process to remove common misspelled forms of the same name such in أحمد and احمد. Table 2 shows the latter case plus the top 20 most frequent male names obtained from the combined Tawjihi and Birzeit lists. We can notice that there is some repetition though the list is supposed to hold distinct names. Item 3 "أحمد" and item 8 "احمد" both represent the same name though one is with "Hamza" and other is not, the correct form is with "Hamza" but it seems that people frequently use "احمد" as well (11752 VS 7714) which is a common mistake.

3.2 Female Names Table

Same is done in building female names table, first filtering female names from Birzeit list by selecting all names with gender equal to *female* then adding all female names found in Tawjihi lists plus all multi classification names that appears with female gender in Birzeit list and appears in 1st field in Tawjihi list.

Table 2: Top 20 Male Names

Item	Name	Frequency		Item	Name	Frequency
1	محمد	41280		11	مصطفى	5031
2	محمود	15662		12	موسى	4649
3	أحمد	11752		13	خالد	4199
4	ابرهيم	9287		14	سليمان	4042
5	حسن	8359		15	سعيد	3897
6	علي	8008		16	عبد الله	3893
7	يوسف	7965		17	جمال	3442
8	احمد	7714		18	اسماعيل	3438
9	خليل	5483		19	صالح	3431
10	حسين	5341		20	عمر	3093

Tawjihi females list is based on the process of filtering male names from students 1st names which is explained in section 3.1.

Same as male names table, the female names are processed to only save unique names with a frequency of occurrence. The total number of distinct female names processed is 2633 (without repetition of same form). Of course this doesn't prove the uniqueness of the names; for example the name "ديما" can be written as "ديمه" or "ديمة" too. Table 3 gives the top 20 female names with their occurrence frequency.

Table 3: Top 20 Female Names

Item	Name	Frequency		Item	Name	Frequency
1	ايمان	2177		11	هبة	1178
2	دعاء	2034		12	نداء	1065
3	الاء	1998		13	سماح	1037
4	ولاء	1673		14	روان	1030
5	حنين	1663		15	هديل	1015
6	اسماء	1506		16	مريم	946
7	اسراء	1297		17	حنان	943
8	فداء	1268		18	فاطمة	912
9	ياسمين	1218		19	صابرين	875
10	عبير	1190		20	اماني	871

3.3 Family Names Table

This is a table that holds family names only, since in Birzeit list and Tawjihi list family names can have male names or female names (in some cases), a filtration process is done by first merging all names with a gender equal to *family* in Birzeit list with all 4th field names in Tawjihi lists (which is the family column). Then we subtracted all male/female names (from male/female tables) and ended up with a table that holds names that isn't in male/female tables.

Same as the other two tables, the family names are processed to only save unique names with a frequency of occurrence. The total number of distinct family names processed is 11209 (without repetition of same form). Table

4 gives the top 20 family names (that only occurs as family names) with their occurrence frequency.

Table 4: Top 20 Family Names

Item	Name	Frequency		Item	Name	Frequency
1	تكروري	952		11	مصري	268
2	حلواني	940		12	جرار	208
3	النجار	450		13	حروب	208
4	عاصي	438		14	الشاعر	203
5	دراغمه	356		15	ربايعة	198
6	بشارات	335		16	رجوب	181
7	جرادات	319		17	سويطي	177
8	دويكات	318		18	صلاحات	175
9	المصري	308		19	شويكي	170
10	ابو الرب	280		20	صوافطه	162

3.4 English Translation Names Table

This is a table that holds each Arabic name (unique) and its different translated forms and a frequency counter for each translated form of the name. This table is useful when building Arabic to English name translating tool. Table 5 gives an example of the name محمد and its top 20 forms in English.

Table 5: The Name محمد and its Different English Translated Forms (Top 20)

#	Name	Freq		#	Name	Freq
1	Mohammad	5513		11	Mohmad	8
2	Muhammad	783		12	Moh'd	8
3	Mohammed	181		13	Mohamd	5
4	Mohamad	168		14	Mohmmed	5
5	Mohummad	157		15	Mouhamad	4
6	Mohamed	44		16	Mouhammad	4
7	Mohmmad	20		17	Mhamad	4
8	Mohammd	12		18	Mhammed	3
9	Muhamad	11		19	Mhmmad	3
10	Muhammed	11		20	Mhmmed	3

From table 5 we can see that most of Birzeit University students with the name محمد as first name or father/grandfather or sometimes family name use the format "Mohammad" by around 78.6% (5513/7011 * 100, where 7011 is the sum of all frequencies of all shapes of محمد, not all forms of the name is shown in Table 5). This means we can consider "Mohammad" as the correct (most used) format for the name محمد and use it as the default translation for the name.

3.5 General Names Table

This is a big table that holds all names appearing in Birzeit and Tawjihi lists with unique forms (a mix of the male, female and family tables), and for each name one gender is assigned as well as a frequency of appearance. For example, the user will find the name نورا , نورة and نوره in this table, but will only find one entry for the name نور with one gender only, so how to calculate the gender in the case of multi classification name? If a name can hold more than one classification/gender then the frequencies of occurrences in all classifications are summed and given to the classification with the highest frequency. In our example the name نور can be used as a male name and a female name. In the original list نور as a male name holds a frequency of 32 and نور as a female name has a frequency of 847. Multi classification flag is added where needed to give multi classification indication about the name.

This table will be used to build an enhanced table for spelling and auto suggestion; the enhanced table will be based on our filtration process (The output of the filtration process); check section 4.1 and 4.2 for the details about building the enhanced names table.

4 NAMES TABLES FILTRATION

After building all these tables a filtration mechanism is applied to detect names with different types of errors that might occur in a naming system such as the common misspelling or writing names in different forms, some are user knowledge errors for example the name ديما has two other forms ديمة and ديمه where the error occurs in the last letter; other name such as آية is more complex since people might do a mistake in the first and the last letters, other results are due to errors in data entry, and so on, we will discuss two type of errors that we handled.

4.1 Names with Different Forms

Fixing common misspelling\errors is looking for the best form to represent a name, and since we are working on a statistical approach, our underlying strategy is to use frequency of appearance of the name in its different formats as the deciding factor, at least in the absence of information against that. So if the name is احمد we must store it as أحمد because the frequency of the latter is the highest. So we are using a statistical approach to judge which is better to use (أحمد، احمد، إحمد) and our male names table says that أحمد has the highest frequency of all, then the frequency of occurrence of احمد and إحمد are summed and added to the frequency of أحمد and then أحمد is considered to be the correct format. We also use Levenshtein distance[4] to give a prejudgment about correct format in a group of names that have a common error between them. Thus the next step was to build groups of names (A group is a list of names with one letter difference), then we used the common errors letters (أ،ا،إ،آ،و،ي،ه،ة) to find which name in each group must be kept and which must be removed. For example assuming a group of (ديمه، ديما، ريمه، ديمة، سيمه) and the group key ديمه we can judge that سيمه and ريمه can be dropped since they differ in a letter that is not in the common errors group, and thus we will end up having ديمة، ديمه، ديما then looking on their frequencies we select ديما, for having the highest frequency among ديمة، ديمه، ديما. However, one can argue that a name like اسامه differs by 2 from the name أسامة and

won't be in the same group. To fix that we rejoined groups that have common elements with potential common errors. Table 6 demonstrates some examples on the filtration process. One can notice that the name اسامه has a group of four shapes after joining two smaller groups (اسامه، اسامة، **أسامه**) and (أسامة، **أسامه**). We notice that أسامة has the largest frequency of appearance which is 431, so the final result will be the sum of all frequencies (the sum is 1151) and the adaption of أسامة as the correct form. Another example with larger list of common errors for the same name is the name آية (Check Table 6).

Table 6: An Example of Names Groups and Filtration

#	Name	Counter	#	Name	Counter
1	أسامة	431	1	أيه	6
2	اسامة	375	2	أية	17
3	أسامه	57	3	آيه	320
4	اسامه	288	4	اية	219
			5	ايه	150
			6	آية	478
Final	أسامة	1151	Final	آية	1190

Also some misspelling may be found in the middle of the name such as مؤيد some may write it موید this is also fixed using the same method described above.

After doing the filtration for all groups we end up with a new table that has the names occurring only once and with a classification converge to the one with highest frequency of occurrences. This table is mainly used in classification/gender detection (Male, Female & Family), also it's useful in autosuggestion of names (section 5.4), and the table was named "*Enhanced Names Table*".

4.2 Compound Names Errors

Another type of error has to do with compound names such as عبد الله، عبد الرحمن، نور الدين. The issue here is whether a space exists between the components or not, which may result in the system treating the single compound name as two simple names. We filter all compound names without space and combine them with same names with space. The *compound names process* looks first for all names with a space element, stores them in a temporary table, then the process loops on all names in the enhanced table and only considers names at one character difference from one (any) of the names stored in the temporary table. If X is a name in the temporary table, then the process looks for names with a difference equal to 1 from X, say Y. If Y differs from X in a space then the frequency of the name with no space character is summed to the frequency of that name with a space character (check Table 7). The process is done after names with different forms filtration is done, thus this is done on the new enhanced names table.

Table 7: Complex names with no space filtration

#	Name	Counter
1	عبد الله	3893
2	عبدالله	1794
Final	عبد الله	5687

5 NAMES METHODS AND TOOLS

The main purpose of this paper is to introduce and build Arabic processing tools specialized in processing Arabic people names. Specifically we are talking about tools like: *Name Gender Detector tool* which tries to detect the classification of an input name (Male, Female or Family), *Names Translation tool* which focuses on translating input Arabic names to their best English equivalent based on statistical considerations, *Names Correction tool* which detects spelling errors in input names and suggesting a correction which can go as deep as needed in finding the best correction for a misspelled name (whether the name is male, female or family), *Names Auto Suggestion tool* which detects and suggests proper names to users while typing queries including names (helpful in the case of search engines and query processing) and *Names extractor tool* which applies the tables processed earlier with their statistics to build an Arabic names extraction tool that extract full and single Arabic names from any text. The description of these tools is given in the coming subsections, also examples (screen shots) of using the above tools can be found in the Appendix at the end of the paper.

5.1 Name Gender Detector Tool

The Name gender detector is a tool that detects the classification of an input name (Male, Female or Family). For example the name محمد is most likely a male name, no females use this name, however in some countries it may be used as a family name. The gender detector tool tries to detect the classification of the name even if it might hold more than one gender (like نور).

To build such a tool, at least two different approaches are possible. One considers some language rules, for example a rule might consider each name ending with (ة) to be a female name. However one needs to define exceptions such as أسامة and most of female names don't end with تاء مربوطة such as روان, ريم, هدى, so it might be difficult to assign and build language rules to confirm the name classification. The other approach adopted here is to base the decisions on statistical data and use the lookup of tables with names with their known gender and frequencies. The enhanced names table comes handy here. The names detector tool receives the name from the user, issues a query to check the existence of the name in the table, if it exists it returns the gender and its percentage of the whole names lists. If not, it returns a null statement with no results found, and the tool push the input to the correction tool (discussed in section 5.3) to check whether the "not found" result

happened due to spelling/common error or not. If it is due to a spelling/common error, the name will be respelled and checked again, else "no result found" will be shown.

An implementation of the testing tool is available online [*http://www.wojoodapis.com/*] and can be easily integrated with any web application. The tool can be customized (using the multi classification flag mentioned in section 3.5) depending on the location of use. That is, if the tool is used to detect male names only, it can consider names such as نور to be male name, despite the high frequency of female classification, because the flag indicate a possibility of multi classification use, same said for names such as ضياء، جهاد، سهام and so on.

5.2 Names Translation Tool

What is meant by names translation is to convert a name to its correct (or widely accepted) translation in English. Some Arabic names have different equivalent English forms (check Table 8). The Translation tool searches in the English translation table and builds a temporary table that holds all possible translations of the input with their frequencies. The forms are sorted in descending order, and then the form with highest frequency is selected as first possible output. In practice the tool may add a suggestion system that offers a second, third … etc best translation if the user did see that returned translation is deemed wrong. Examples using the tool are found in Table 8.

Table 8: Names Samples and Different English Translated Forms

#	Arabic Name	English Translation	Freq	#	Arabic Name	English Translation	Freq
1	سمير	Samir	299	4	أحمد	Ahmad	1875
		Sameer	85			Ahmed	48
2	نورا	Noura	19			Ahamad	6
		Nora	7	5	مؤيد	Mo'ayad	10
		Nura	5			Mu'ayad	9
		Noora	3			Moayad	5
3	رياض	Riyad	148			Mu'ayyad	5
		Riad	24			Mo'ayyad	3
		Reyad	8			Muayad	3

5.3 Names Correction Tool

The Name correction tool is used to correct people names. It has the flavour of a general spell checker but with a smaller dictionary (people names based dictionary instead of all words dictionary). At first any entered name is checked against the enhanced names table. If there is a match, the name is considered correct; otherwise a spellchecking mechanism is invoked to get the best group of names that can replace the entered name. This correction tool was built on previous work we did on building a correction algorithm to enhance Arabic web search quires[5].

In this subsection we briefly explain how our general algorithm works taking into consideration that we are using a smaller dictionary. Mainly, the correction algorithm depends on two major components, *Levenshtein distance* which works as a metric that gives the minimum number of steps needed to convert string A to string(s) B[4]; string A in our case is the input miss spelled name and string(s) B is the possible correct name(s) that A can be converted to, (B represent a name(s) from the enhanced names table) and a *ranking system* that takes different measurements in consideration to sort possible correct outputs to misspelled input.

Levenshtein distance is first calculated between the entered name and each name in the dictionary. Only dictionary names with minimum differences are considered (taking the names with the minimal distance from the entered name), a list of these names is stored and then the ranking system is used to detect which is the best fit to the entered name. Realizing that errors may result from several factors including exchanging letters with similar shapes such in the case of سمير، شمير (Shape Similarity), using adjacent keys in place of the intended letter (Letter Location) and typing a similar sounding character such as سمير، صمير (Soundex[6]). The ranking system takes each of these factors into consideration in the search for a replacement. Equation (1) applies in ranking when ordering the possible outputs (More details about the correction algorithm and ranking system can be found in a previous work[5]).

$$\text{Rank}(W) = A*\text{Freq} + B*\text{Similarity} + C*\text{Location} + D*\text{Soundex} \quad (1)$$

Where A, B, C, D are percentages with summation of 100% (weights). Considering A = 0.5, B = 0.20, C= 0.25 and D = 0.05. The chosen values for A, B, C and D are not necessarily the best and in future work more testing based on experimentation will be done to decide the best range (or values) for them. Check Table 9 for some examples.

Current tests consist of testing lists of misspelled names and comparing the output results with a predefined correct list of the same names. Two types of testing lists were made, an auto generated lists (the lists were created by randomly changing the names by adding, swapping, replacing and deleting 1, 2, and 3 letters) and a speed typing based list (a list that was generated by manually speed typing without looking at the keyboard). Each input list has 100 elements and the results (1st, 2nd and 3rd best outputs of the correction algorithm) were compared with the true original correct inputs, the results are demonstrated in Table 10.

5.4 Names Auto Suggestion Tool

This is a general autosuggestion for names which can be used in different applications where name entry is needed; it suggests names while typing. The main challenge here lies in suggesting the form with the highest frequency of occurrence and thus has more likelihood of being the intended entry.

As explained in section 4.1 we only adopt one form for every name in our enhanced names table which we consider correct. Common spelling errors of names can be divided

into four categories: common first letter misspelling, common middle letter misspelling, last letter misspelling and complex names spelling errors. And since we know the possible typing errors and have general and enhanced names tables, we can use this knowledge to user input while typing.

Table 9: Names Correction Test Sample

#	Input	Output(s)		Input	Output(s)
1	ديم	ريم، ديما،كيم، نديم	5	اية	راية، أية
2	شوشن	سوسن، شوكت، سوزان، روان	6	نوزالدين	نور الدين
3	خاقلين	تالين، جاكلين، مارلين، كاتلين، مادلين	7	رمري	رمزي، رازي
4	اقراجيم	إبراهيم	8	غبير	عبير، غدير

Table 10: Early Tests on Names Correction

#	Test Type	Pass Percentage[a]
1	Speed Writing (test1)	87%
	Speed Writing (test2)	84%
	Speed Writing (test3)	85%
2	Auto generated errors	
	- One Error	91%
	- Two Errors	79%
	- Three Errors	70%

a. These percentages are a subject of change by future tests and improvements.

The challenge is that users might start typing incorrectly, for example, a user enters أحمد but starts with ا not أ and thus will never end up detecting أحمد as a possible completion if only the enhanced table is used (because we only have one form which is أحمد) . So a modification on user input is needed in runtime and the tool will automatically take the possibility of changing the first letter (in case of ا to أ or آ or إ) and then wait for the next letter. The same is said in case of middle letter, مؤيد for example the user might enter the name as موید and the tool will take the possibility of ؤ while typing. Last letter case is different since the autosuggestion tool will be able to detect the correct form and use it based on all previous entered letters.

5.5 Names Extraction Tool

Names extraction is a method to extract people names from Arabic text, the method mainly depends on a merged table from male, female and family tables, this table has names with their gender and frequency but with no filtration (the general names table), which means that one can find the name أحمد and the name احمد both with their frequencies of appearance. This is because in name extraction we should not assume the input text has names in the formats that we think are the most used. For example it's not uncommon to find credible Arabic text with multiple spellings of أحمد. The extraction function first parses all the input Arabic text comparing each word with the general names table entries. If the table has the word (which means the current processed word is probably a name) then the function directly parse and check three words ahead (word+1, word+2, word+3) in order to decide if the extracted name is single or is a full name (a name appears with other name(s) that indicate a father, grandfather or family name), if the name is followed by another name(s) then the full series of names is compared with predefined names types (<male[0],male[1],…male[i] ‖ family>, <female, male[1],…male[i] ‖ family>) if the series match one of the names types, then it is considered a full name, else partial parsing is done to detect any full names series occurring inside the current series (to split single names from full names), for example the series رامي محمد حمدان matches <male, male, family> however the series رامي هند محمد حمدان doesn't match a series, however a partial series هند محمد حمدان matches <female, male, family> and thus is considered a full name and the name رامي is considered a single name. However not every single name found is considered a name. For example the name جميل might be found in a text but as an adjective not as a person name. We didn't try to use grammar rules here but only defined and applied some rules that we followed to decide if a single name is a person name or not: to consider a word to be a single name (when it can hold more than one type of definition) the following rules are applied: it either appears more than N times in the text (currently N = 3) or appears in a full name in the text, for example جميلة سمير النتشة then جميلة and its other form (جميله) and سمير if found single in the same text will be detected; and finally if it appears in series such as علي و ذكي و سامي و انس ذهبوا إلى الجامعة or appears after a defining term (such as السيد، الدكتور، الآنسة ... etc) .

6 CONCLUSIONS

We presented some useful tools that can help processing people names in digital documents and web content. Our work aims to design query/forms pre-processing name tools that are able to efficiently process and identify Arabic people names in search queries and digital documents. We addressed how people names resources were collected and processed to build a statistical/database-based approach, based on database statistics we constructed databases of Arabic names and their frequencies, different types of tables were built and used as an infrastructure for the well structured people names tools which are capable of being integrated into existing web search engines and document processing systems, tools such as : name gender detector, translation (Arabic to English), correction, auto suggestion and names extraction. It is important to note that while much of the decision making is dictated by usage statistics and the approach calls for no deep understanding of name writing rules, processing tables' data by experts to make sure that the selected formats are in line with language rules can only improve performance.

Future work will consider recommendations from Arabic language specialists and experts about the correct forms of names in order to match them with our statistical databases

and improve the quality of our results. Also we will consider the problem of extraction words (that can be people names) with multiple meanings (Polysemy) which addresses word-sense disambiguation (WSD), an open problem in natural language processing that tries to figure the sense of a word that is being used in a sentence.

REFERENCES

[1] Supreme Council of Information & Comminication Technology, Arab Digital Content, ictQATAR Website.[Online].Available: www.ictqatar.qa/output/Page2039.asp/,[Mar, 20,2011].

[2] Mohammed bin Rashid Al Maktoum Foundation, Sawaed Programme, Mohammed bin Rashid Al Maktoum Foundation Website.[Online].Available: http://www.mbrfoundation.ae/English/Entrepreneurship/Pages/Sawaed.aspx /,[Mar, 20,2011].

[3] ESCWA, Digital Arabic Content, ESCWA Website.[Online].Available: http://www.escwa.un.org/divisions/projects/dac/index.asp/,[Mar, 20,2011].

[4] Wikipedia, Levenshtein Distance, Wikipedia Website.[Online].Available: http://en.wikipedia.org/wiki/Levenshtein_distance/,[Mar, 5,2011].

[5] Adnan Yahya, Ali Salhi: " Enhancement Tools for Arabic Web Search : A Statistical Approach"; *7th International Conference on Innovations in Information Technology*; Abu Dhabi, United Arab Emirates.; 25-27 April 2011. Website.[Online].Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5893871&isnumber=5893793

[6] Wikipedia, Soundex, Wikipedia Website.[Online].Available: http://en.wikipedia.org/wiki/Soundex,[Mar,5,2011].

APPENDIX

The following figures show some examples of using the names tools (Figure 1-4 show examples of using Names Correction, Gender Detector, Names Auto Suggestion and Names Translating tools. Figure 5 shows example of using the Names Extraction tool). For more demos and online test please visit our website on: *http://www.wojoodapis.com/* , and for names tool testing please visit: *http://www.wojoodapis.com/Test/NamesFillTest.html* and *http://www.wojoodapis.com/findname.html*.

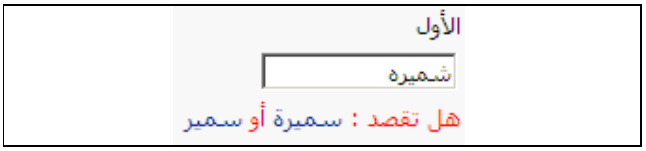

Figure A-1: An example of Names Correction tool. The user enters شميره and the system suggest the use of سميرة with "ة" not "ه" since it's the form with highest statistical use , plus an option of selecting سمير (as second option).

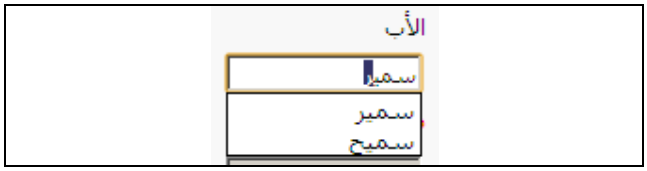

Figure A-2: An example of Name Auto Suggestion tool with Gender Detector tool working together. As noticed the system suggests only male names in the field of father name due of the use of the Gender Detector tool, so the system will not suggest سميرة for example.

Figure A-3: An example of using the Name Auto Suggestion tool; as can be noticed from the result of the suggestion (after entering two letters) has different forms of "Alf" , for example ايمان with no "Hamza" since it's the name's form with highest statistical use and آية with "mada" since it's the form with highest statistical use and same can be said about other names such as أيمن .

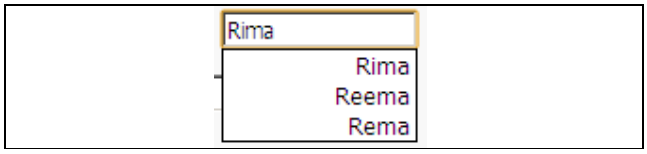

Figure A-4: An example of the Translation tool. The user entered ريما and the tool translated the name to "Rima" , however clicking on the translated name gives options for user to select another translated form if needed.

Figure 5: An example of using the Names Extraction. As can be noticed from the partial screen shot, the name سائد ناشف is detected to be a full name that matches <Male, Family>.