COLABA: Arabic Dialect Annotation and Processing

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, Yassine Benajiba

Center for Computational Learning Systems
475 Riverside Drive, Suite 850
New York, NY 10115
Columbia University
{mdiab,habash,rambow,mtantawy,ybenajiba}@ccls.columbia.edu

Abstract

In this paper, we describe COLABA, a large effort to create resources and processing tools for Dialectal Arabic Blogs. We describe the objectives of the project, the process flow and the interaction between the different components. We briefly describe the manual annotation effort and the resources created. Finally, we sketch how these resources and tools are put together to create DIRA, a term-expansion tool for information retrieval over dialectal Arabic collections using Modern Standard Arabic queries.

1. Introduction

The Arabic language is a collection of historically related variants. Arabic dialects, collectively henceforth Dialectal Arabic (DA), are the day to day vernaculars spoken in the Arab world. They live side by side with Modern Standard Arabic (MSA). As spoken varieties of Arabic, they differ from MSA on all levels of linguistic representation, from phonology, morphology and lexicon to syntax, semantics, and pragmatic language use. The most extreme differences are on phonological and morphological levels.

The language of education in the Arab world is MSA. DA is perceived as a lower form of expression in the Arab world; and therefore, not granted the status of MSA, which has implications on the way DA is used in daily written venues. On the other hand, being the spoken language, the native tongue of millions, DA has earned the status of living languages in linguistic studies, thus we see the emergence of serious efforts to study the patterns and regularities in these linguistic varieties of Arabic (Brustad, 2000; Holes, 2004; Bateson, 1967; Erwin, 1963; Cowell, 1964; Rice and Sa'id, 1979; Abdel-Massih et al., 1979). To date most of these studies have been field studies or theoretical in nature with limited annotated data. In current statistical Natural Language Processing (NLP) there is an inherent need for large-scale annotated resources for a language. For DA, there has been some limited focused efforts (Kilany et al., 2002; Maamouri et al., 2004; Maamouri et al., 2006); however, overall, the absence of large annotated resources continues to create a pronounced bottleneck for processing and building robust tools and applications.

DA is a pervasive form of the Arabic language, especially given the ubiquity of the web. DA is emerging as the language of informal communication online, in emails, blogs, discussion forums, chats, SMS, etc, as they are media that are closer to the spoken form of language. These genres pose significant challenges to NLP in general for any language including English. The challenge arises from the fact that the language is less controlled and more speech like while many of the textually oriented NLP techniques are tailored to processing edited text. The problem is compounded for Arabic precisely because of the use of DA in these genres. In fact, applying NLP tools designed for MSA directly to DA yields significantly lower performance, making it imperative to direct the research to building resources and dedicated tools for DA processing.

DA lacks large amounts of consistent data due to two factors: a lack of orthographic standards for the dialects, and a lack of overall Arabic content on the web, let alone DA content. These lead to a severe deficiency in the availability of computational annotations for DA data. The project presented here – Cross Lingual Arabic Blog Alerts (COLABA) – aims at addressing some of these gaps by building large-scale annotated DA resources as well as DA processing tools.[1]

This paper is organized as follows. Section 2. gives a high level description of the COLABA project and reviews the project objectives. Section 3. discusses the annotated resources being created. Section 4. reviews the tools created for the annotation process as well as for the processing of the content of the DA data. Finally, Section 5. showcases how we are synthesizing the resources and tools created for DA for one targeted application.

2. The COLABA Project

COLABA is a multi-site partnership project. This paper, however, focuses only on the Columbia University contributions to the overall project.

COLABA is an initiative to process Arabic social media data such as blogs, discussion forums, chats, etc. Given that the language of such social media is typically DA, one of the main objective of COLABA is to illustrate the significant impact of the use of dedicated resources for the processing of DA on NLP applications. Accordingly, together with our partners on COLABA, we chose Information Retrieval (IR) as the main testbed application for our ability to process DA.

Given a query in MSA, using the resources and processes created under the COLABA project, the IR system is able to retrieve relevant DA blog data in addition to MSA data/blogs, thus allowing the user access to as much Arabic

[1] We do not address the issue of augmenting Arabic web content in this work.

content (in the inclusive sense of MSA and DA) as possible. The IR system may be viewed as a cross lingual/cross dialectal IR system due to the significant linguistic differences between the dialects and MSA. We do not describe the details of the IR system or evaluate it here; although we allude to it throughout the paper.

There are several crucial components needed in order for this objective to be realized. The COLABA IR system should be able to take an MSA query and convert it/translate it, or its component words to DA or alternatively convert all DA documents in the search collection to MSA before searching on them with the MSA query. In COLABA, we resort to the first solution. Namely, given MSA query terms, we process them and convert them to DA. This is performed using our DIRA system described in Section 5.. DIRA takes in an MSA query term(s) and translates it/(them) to their corresponding equivalent DA terms. In order for DIRA to perform such an operation it requires two resources: a lexicon of MSA-DA term correspondences, and a robust morphological analyzer/generator that can handle the different varieties of Arabic. The process of creating the needed lexicon of term correspondences is described in detail in Section 3.. The morphological analyzer/generator, MAGEAD, is described in detail in Section 4.3..

For evaluation, we need to harvest large amounts of data from the web. We create sets of queries in domains of interest and dialects of interest to COLABA. The URLs generally serve as good indicators of the dialect of a website; however, given the fluidity of the content and variety in dialectal usage in different social media, we decided to perform dialect identification on the lexical level.

Moreover, knowing the dialect of the lexical items in a document helps narrow down the search space in the underlying lexica for the morphological analyzer/generator. Accordingly, we will also describe the process of dialect annotation for the data.

The current focus of the project is on blogs spanning four different dialects: Egyptian (EGY), Iraqi (IRQ), Levantine (LEV), and (a much smaller effort on) Moroccan (MOR). Our focus has been on harvesting blogs covering 3 domains: social issues, religion and politics.

Once the web blog data is harvested as described in Section 3.1., it is subjected to several processes before it is ready to be used with our tools, namely MAGEAD and DIRA. The annotation steps are as follows:

1. **Meta-linguistic Clean Up.** The raw data is cleaned from html mark up, advertisements, spam, encoding issues, and so on. Meta-linguistic information such as date and time of post, poster identity information and such is preserved for use in later stages.

2. **Initial Ranking of the Blogs.** The sheer amount of data harvested is huge; therefore, we need to select blogs that have the most dialectal content so as to maximally address the gap between MSA and DA resources. To that end, we apply a simple DA identification (DI) pipeline to the blog document collection ranking them by the level of dialectal content. The DI pipeline is described in detail in Section 4.2.. The intuition is that the more words in the blogs that are not analyzed or recognized by a MSA morphological analyzer, the more dialectal the blog. It is worth noting that at this point we only identify that words are not MSA and we make the simplifying assumption that they are DA. This process results in an initial ranking of the blog data in terms of dialectness.

3. **Content Clean-Up.** The content of the highly ranked dialectal blogs is sent for an initial round of manual clean up handling speech effects and typographical errors (typos) (see Section3.2.). Additionally, one of the challenging aspects of processing blog data is the severe lack of punctuation. Hence, we add a step for sentence boundary insertion as part of the cleaning up process (see Section 3.3.). The full guidelines will be presented in a future publication.

4. **Second Ranking of Blogs and Dialectalness Detection**. The resulting cleaned up blogs are passed through the DI pipeline again. However, this time, we need to identify the actual lexical items and add them to our lexical resources with their relevant information. In this stage, in addition to identifying the dialectal unigrams using the DI pipeline as described in step 2, we identify out of vocabulary bigrams and trigrams allowing us to add entries to our created resources for words that look like MSA words (i.e. cognates and faux amis that already exist in our lexica, yet are specified only as MSA). This process renders a second ranking for the blog documents and allows us to hone in on the most dialectal words in an efficient manner. This process is further elaborated in Section 4.2..

5. **Content Annotation**. The content of the blogs that are most dialectal are sent for further content annotation. The highest ranking blogs undergo full word-by-word dialect annotation as described in Section 3.5.. Based on step 4, the most frequent surface words that are deemed dialectal are added to our underlying lexical resources. Adding an entry to our resources entails rendering it in its lemma form since our lexical database uses lemmas as its entry forms. We create the underlying lemma (process described in Section 3.6.) and its associated morphological details as described in Section 3.7.. Crucially, we tailor the morphological information to the needs of MAGEAD. The choice of surface words to be annotated is ranked based on the word's frequency and its absence from the MSA resources. Hence the surface forms are ranked as follows: unknown frequent words, unknown words, then known words that participate in infrequent bigrams/trigrams compared to MSA bigrams/trigrams. All the DA data is rendered into a Colaba Conventional Orthography (CCO) described in Section 3.4.. Annotators are required to use the CCO for all their content annotations.

To efficiently clean up the harvested data and annotate its content, we needed to create an easy to use user interface with an underlying complex database repository that organizes the data and makes it readily available for further research. The annotation tool is described in Section 4.1..

3. Resource Creation

Resource creation for COLABA is semi automatic. As mentioned earlier, there is a need for a large collection of data to test out the COLABA IR system. The data would ideally have a large collection of blogs in the different relevant dialects in the domains of interest, annotated with the relevant levels of linguistic knowledge such as degree of dialectness and a lexicon that has coverage of the lexical items in the collection. Accordingly, the blog data is harvested using a set of identified URLs as well as queries that are geared towards the domains of interest in the dialect.

3.1. Data Harvesting

Apart from identifying a set of URLs in each of the relevant dialects, we designed a set of DA queries per dialect to harvest large quantities of DA data from the web. These queries were generated by our annotators with no restrictions on orthographies, in fact, we gave the explicit request that they provide multiple variable alternative orthographies where possible. The different dialects come with their unique challenges due to regional variations which impact the way people would orthographically represent different pronunciations. For example, DA words with MSA cognates whose written form contains the ق *q*[2] (*Qaf*) consonant may be spelled etymologically (as ق *q*) or phonologically as one of many local variants: ك *k*, أ *Â* or گ *G*.

We collected 40 dialectal queries from each of our 25 annotators specifically asking them when possible to identify further regional variations. In our annotations in general, we make the gross simplifying assumption that Levantine (Syrian, Lebanese, Palestinian and Jordanian) Arabic is a single dialect. However, for the process of query generation, we asked annotators to identify sub-dialects. So some of our queries are explicitly marked as Levantine-Palestinian or Levantine-Syrian for instance. Moreover, we asked the annotators to provide queries that have verbs where possible. We also asked them to focus on queries related to the three domains of interest: politics, religion and social issues. All queries were generated in DA using Arabic script, bearing in mind the lack of orthographic standards. The annotators were also asked to provide an MSA translation equivalent for the query and an English translation equivalent. Table 1 illustrates some of the queries generated.

3.2. Typographical Clean Up

Blog data is known to be a challenging genre for any language from a textual NLP perspective since it is more akin to spoken language. Spelling errors in MSA (when used) abound in such genres which include speech effects. The problem is compounded for Arabic since there are no DA orthographic standards. Accordingly, devising guidelines for such a task is not straight forward. Thus, we simplified the task to the narrow identification of the following categories:

- MSA with non-standard orthography, e.g., هذة *hðħ* 'this' becomes هذه *hðh*, and المساجذ *AlmsAjð* 'mosques' becomes المساجد *AlmsAjd*.
- Speech Effects (SE) are typical elongation we see in blog data used for emphasis such as كووووره *kwwwwrh* 'ball' is rendered كورة *kwrħ*.
- Missing/Added Spaces (MS/AS) are cases where there is obviously a missing space between two or more words that should have been rendered with a space. For example, in EGY, متكلشالبرتأنة *mtklšAlbrtÂnħ* 'don't eat the orange' is turned into متكلش البرتأنة *mtklš AlbrtÂnħ*. Note that in this dialectal example, we do not require the annotator to render the word for orange البرتأنة *AlbrtÂnħ* in its MSA form, namely, البرتقالة *AlbrtqAlħ*.

3.3. Sentence Boundary Detection

In blogs, sentence boundaries are often not marked explicitly with punctuation. In this task, annotators are required to insert boundaries between sentences. We define a sentence in our guidelines as a syntactically and semantically coherent unit in language. Every sentence has to have at least a main predicate that makes up a main clause. The predicate could be a verb, or in the case of verb-less sentences, the predicate could be a nominal, adjectival or a prepositional phrase. Table 2 illustrates a blog excerpt as it occurs naturally on the web followed by sentence boundaries explicitly inserted with a carriage return splitting the line in three sentences.

3.4. COLABA Conventional Orthography

Orthography is a way of writing language using letters and symbols. MSA has a standard orthography using the Arabic script. Arabic dialects, on the other hand, do not have a standard orthographic system. As such, a variety of approximations (phonological/lexical/etymological) are often pursued; and they are applied using Arabic script as well as Roman/other scripts. In an attempt to conventionalize the orthography, we define a phonological scheme which we refer to as the COLABA Conventional Orthography (CCO). This convention is faithful to the dialectal pronunciation as much as possible regardless of the way a word is typically written. This scheme preserves and explicitly represents all the sounds in the word including the vowels. For example, باب *bAb* 'door' is rendered as *be:b* in CCO for LEV (specifically Lebanese) but as *ba:b* for EGY.[3] The full guidelines will be detailed in a future publication.

[2] All Arabic transliterations are provided in the Habash-Soudi-Buckwalter (HSB) transliteration scheme (Habash et al., 2007).

[3] Most CCO symbols have English-like/HSB-like values, e.g., ب *b* or م *m*. Exceptions include *T* (ث *θ*), *D* (ذ ð), *c* (ش š), *R* (غ *γ*), *7* (ح *H*), *3* (ع ς), and *2* (ء ’). CCO uses '.' to indicate emphasis/velarization, e.g., *t.* (ط *T*).

DA Query	DA	MSA	English
الطلاق بقى ظاهره	EGY	الطلاق اصبح ظاهرة	divorce became very common
راح احجيلكم	IRQ	سوف اروي لكم	I will tell you a story
راح دوز ع قبر بيه	LEV	ذهب فورا الى قبر ابيه	He went directly to visit his father’s tomb
ما زال شاد فراسو	MOR	لا زال في وضع جيد	he is still in good shape

Table 1: Sample DA queries used for harvesting blog data

Input text
بدي اخذ ماجيستير ودكتوراه بدي اتزوج وجيب ولاد وبدي عيش بجو اسري كله مودة
After manual sentence boundary detection
بدي اخذ ماجيستير ودكتوراه بدي اتزوج وجيب ولاد وبدي عيش بجو اسري كله مودة

Table 2: LEV blog excerpt with sentence boundaries identified.

- CCO explicitly indicates the pronounced short vowels and consonant doubling, which are expressed in Arabic script with optional diacritics. Accordingly, there is no explicit marking for the sukuun diacritic which we find in Arabic script. For example, the CCO for مركب *mrkb* in EGY could be *markib* ‘boat’ or *mirakkib* ‘something put together/causing to ride’ or *murakkab* ‘complex’.

- Clitic boundaries are marked with a +. This is an attempt at bridging the gap between phonology and morphology. We consider the following affixations as clitics: conjunctions, prepositions, future particles, progressive particles, negative particles, definite articles, negative circumfixes, and attached pronouns. For example, in EGY CCO وسلام *wslAm* ‘and peace’ is rendered *we+sala:m* and مايكتبش *mAyktbš* ‘he doesn’t write’ is rendered *ma+yiktib+c*.

- We use the ^ symbol to indicate the presence of the Ta Marbuta (feminine marker) morpheme or of the Tanween (nunation) morpheme (marker of indefiniteness). For example, مكتبة *mktbħ* ‘library’ is rendered in CCO as *maktaba*^ (EGY). Another example is عمليًا *ςmlyAã* ‘practically’, which is rendered in CCO as *3amaliyyan*^.

CCO is comparable to previous efforts on creating resources for Arabic dialects (Maamouri et al., 2004; Kilany et al., 2002). However, unlike Maamouri et al. (2004), CCO is not defined as an Arabic script dialectal orthography. CCO is in the middle between the morphophonemic and phonetic representations used in Kilany et al. (2002) for Egyptian Arabic. CCO is quite different from commonly used transliteration schemes for Arabic in NLP such as Buckwalter transliteration in that CCO (unlike Buckwalter) is not bijective with Arabic standard orthography.
For the rest of this section, we will use CCO in place of the HSB transliteration except when indicated.

3.5. Dialect Annotation

Our goal is to annotate all the words in running text with their degree of dialectalness. In our conception, for the purposes of COLABA we think of MSA as a variant dialect; hence, we take it to be the default case for the Arabic words in the blogs. We define a dialectal scale with respect to orthography, morphology and lexicon. We do not handle phrasal level or segment level annotation at this stage of our annotation, we strictly abide by a word level annotation.[4] The annotators are required to provide the CCO representation (in Section 3.4.) for all the words in the blog. If a word as it appears in the original blog maintains its meaning and orthography as in MSA then it is considered the default MSA for dialect annotation purposes, however if it is pronounced in its context dialectically then its CCO representation will reflect the dialectal pronunciation, e.g. يكتب, *yktb* ‘he writes’ is considered MSA from a dialect annotation perspective, but in an EGY context its CCO representation is rendered *yiktib* rather than the MSA CCO of *yaktub*.
Word dialectness is annotated according to a 5-point scale building on previous efforts by Habash et al. (2008):

- WL1: MSA with dialect morphology بيكتب *bi+yiktib* ‘he is writing’, هيكتب *ha+yiktib* ‘he will write’

- WL2: MSA faux amis where the words look MSA but are semantically used dialectically such as عم *3am* a LEV progressive particle meaning ‘in the state of’ or MSA ‘uncle’

- WL3: Dialect lexeme with MSA morphology such as سيزعل *sa+yiz3al* ‘he will be upset’

- WL4: Dialect lexeme where the word is simply a dialectal word such as the negative particle مش *mic* ‘not’

[4] Annotators are aware of multiword expressions and they note them when encountered.

- WL5: Dialect lexeme with a consistent systematic phonological variation from MSA, e.g., LEV تلاتة *tala:te*^ 'three' versus ثلاثة *Tala:Ta*^.

In addition, we specify another six word categories that are of relevance to the annotation task on the word level: Foreign Word (جيلاتو, *jila:to*, 'gelato ice cream'), Borrowed Word (ويك اند, *wi:k 2end*, 'weekend'), Arabic Named Entity (عمرو دياب, *3amr dya:b*, 'Amr Diab'), Foreign Named Entity (جيمي كارتر, *jimi kartar*, 'Jimmy Carter'), Typo (further typographical errors that are not caught in the first round of manual clean-up), and in case they don't know the word, they are instructed to annotate it as unknown.

3.6. Lemma Creation

This task is performed for a subset of the words in the blogs. We focus our efforts first on the cases where an MSA morphological analyzer fails at rendering any analysis for a given word in a blog. We are aware that our sampling ignores the faux amis cases with MSA as described in Section 3.5.. Thus, for each chosen/sampled dialectal surface word used in an example usage from the blog, the annotator is required to provide a lemma, an MSA equivalent, an English equivalent, and a dialect ID. All the dialectal entries are expected to be entered in the CCO schema as defined in Section 3.4..

We define a lemma (citation form) as the basic entry form of a word into a lexical resource. The lemma represents the semantic core, the most important part of the word that carries its meaning. In case of nouns and adjectives, the lemma is the definite masculine singular form (without the explicit definite article). And in case of verbs, the lemma is the 3rd person masculine singular perfective active voice. All lemmas are clitic-free.

A dialectal surface word may have multiple underlying lemmas depending on the example usages we present to the annotators. For example, the word مركبه *mrkbh* occurs in two examples in our data: 1. سامي مركبه بإيديه *sa:mi mirakkib+uh be+2ide:+h* 'Sami built it with his own hands' has the corresponding EGY lemma *mirakkib* 'build'; and 2. الرجالة راحوا يشتروا مركبه منه *ir+rigga:la*^ *ra:7u yictiru markib+uh minn+uh* 'The men went to buy his boat from him' with the corresponding lemma *markib* 'boat'. The annotators are asked to explicitly associate each of the created lemmas with one or more of the presented corresponding usage examples.

3.7. Morphological Profile Creation

Finally, we further define a morphological profile for the entered lemmas created in Section 3.6.. A computationally oriented morphological profile is needed to complete the necessary tools relevant for the morphological analyzer MAGEAD (see Section 4.3.). We ask the annotators to select (they are given a list of choices) the relevant part-of-speech tag (POS) for a given lemma as it is used in the blogs. For some of the POS tags, the annotators are requested to provide further morphological specifications.

In our guidelines, we define coarse level POS tags by providing the annotators with detailed diagnostics on how to identify the various POS based on form, meaning, and grammatical function illustrated using numerous examples. The set of POS tags are as follows: (Common) Noun, Proper Noun, Adjective, Verb, Adverb, Pronoun, Preposition, Demonstrative, Interrogative, Number, and Quantifier. We require the annotators to provide a detailed morphological profile for three of the POS tags mentioned above: Verb, Noun and Adjective. For this task, our main goal is to identify irregular morphological behavior. They transcribe all their data entries in the CCO representation only as defined in Section 3.4.. We use the Arabic script below mainly for illustration in the following examples.

- **Verb Lemma:** In addition to the basic 3rd person masculine singular (3MS) active perfective form of the dialectal verb lemma, e.g., شرب *cirib* 'he drank' (EGY), the annotators are required to enter: (i) the 3MS active imperfective يشرب *yicrab*; (ii) the 3MS passive perfective is انشرب *incarab*; (iii) the 3MS passive imperfective ينشرب *yincirib*; and (iv) and the masculine singular imperative اشرب *icrab*.

- **Noun Lemma:** The annotators are required to enter the feminine singular form of the noun if available. They are explicitly asked not to veer too much away from the morphological form of the lemma, so for example, they are not supposed to put ست *sit* 'woman/lady' as the feminine form of راجل *ra:gil* 'man'. The annotators are asked to specify the rationality/humanness of the noun which interacts in Arabic with morphosyntactic agreement. Additional optional word forms to provide are any broken plurals, mass count plural collectives, and plurals of plurals, e.g *rigga:la*^ and *riga:l* 'men' are both broken plurals of *ra:gil* 'man'.

- **Adjective Lemma:** For adjectives, the annotators provide the feminine singular form and any broken plurals, e.g. the adjective أول *2awwel* 'first [masc.sing]' has the feminine singular form أولى *2u:la* and the broken plural أوائل *2awa:2il*.

4. Tools for COLABA

In order to process and manage the large amounts of data at hand, we needed to create a set of tools to streamline the annotation process, prioritize the harvested data for manual annotation, then use the created resources for MAGEAD.

4.1. Annotation Interface

Our annotation interface serves as the portal which annotators use to annotate the data. It also serves as the repository for the data, the annotations and management of the annotators. The annotation interface application runs on a web server because it is the easiest and most efficient way to allow different annotators to work remotely, by entering their annotations into a central database. It also manages the annotators tasks and tracks their activities efficiently. For a more detailed description of the interface see (Benajiba and

Diab, 2010). For efficiency and security purposes, the annotation application uses two different servers. In the first one, we allocate all the html files and dynamic web pages. We use PHP to handle the dynamic part of the application which includes the interaction with the database. The second server is a database server that runs on PostgreSQL.[5] Our database comprises 22 relational databases that are categorized into tables for:

- Basic information that is necessary for different modules of the application. These tables are also significantly useful to ease the maintenance and update of the application.
- User permissions: We have various types of users with different permissions and associated privileges. These tables allow the application to easily check the permissions of a user for every possible action.
- Annotation information: This is the core table category of our database. Its tables save the annotation information entered by each annotator. They also save additional information such as the amount of time taken by an annotator to finish an annotation task.

For our application, we define three types of users, hence three views (see Figure 1):

1. *Annotator.* An Annotator can perform an annotation task, check the number of his/her completed annotations, and compare his/her speed and efficiency against other annotators. An annotator can only work on one dialect by definition since they are required to possess native knowledge it. An annotator might be involved in more than one annotation task.
2. *Lead Annotator.* A Lead annotator (i) manages the annotators' accounts, (ii) assigns a number of task units to the annotators, and, (iii) checks the speed and work quality of the annotators. Leads also do the tasks themselves creating a gold annotation for comparison purposes among the annotations carried out by the annotators. A lead is an expert in only one dialect and thus s/he can only intervene for the annotations related to that dialect.
3. *Administrator.* An Administrator (i) manages the Leads' accounts, (ii) manages the annotators' accounts, (iii) transfers the data from text files to the database, (iv) purges the annotated data from the data base to xml files, and (v) produces reports such as inter-annotator agreement statistics, number of blogs annotated, etc.

The website uses modern JavaScript libraries in order to provide highly dynamic graphical user interfaces (GUI). Such GUIs facilitate the annotator's job leading to significant gain in performance speed by (i) maximizing the number of annotations that can be performed by a mouse click rather than a keyboard entry and by (ii) using color coding for fast checks. Each of the GUIs which compose our web applications has been carefully checked to be consistent with the annotation guidelines.

4.2. DA Identification Pipeline

We developed a simple module to determine the degree to which a text includes DA words. Specifically, given Arabic text as input, we were interested in determining how many words are not MSA. The main idea is to use an MSA morphological analyzer, Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004), to analyze the input text. If BAMA is able to generate a morphological analysis for an input word, then we consider that word MSA.

As a result, we have a conservative assessment of the dialectness of an input text. A major source of potential errors are names which are not in BAMA.

We assessed our pipeline on sample blog posts from our harvested data. In an EGY blog post[6] 19% of the word types failed BAMA analysis. These words are mainly DA words with few named entities. Similar experiments were conducted on IRQ,[7] LEV,[8] and MOR[9] blog posts yielding 13.5%, 8% and 26% of non-MSA word types, respectively. It is worth noting the high percentage of out of vocabulary words for the Moroccan thread compared to the other dialects. Also, by comparison, the low number of misses for Levantine. This may be attributed to the fact that BAMA covers some Levantine words due to the LDC's effort on the Levantine Treebank (Maamouri et al., 2006).

We further analyzed BAMA-missed word types from a 30K word blog collection. We took a sample of 100 words from the 2,036 missed words. We found that 35% are dialectal words and that 30% are named entities. The rest are MSA word that are handled by BAMA. We further analyzed two 100 string samples of least frequent bigrams and trigrams of word types (measured against an MSA language model) in the 30K word collection. We found that 50% of all bigrams and 25% of trigrams involved at least one dialectal word. The percentages of named entities for bigrams and trigrams in our sample sets are 19% and 43%, respectively.

4.3. MAGEAD

MAGEAD is a morphological analyzer and generator for the Arabic language family, by which we mean both MSA and DA. For a fuller discussion of MAGEAD (including an evaluation), see (Habash et al., 2005; Habash and Rambow, 2006; Altantawy et al., 2010). For an excellent discussion of related work, see (Al-Sughaiyer and Al-Kharashi, 2004). MAGEAD relates (bidirectionally) a lexeme and a set of linguistic features to a surface word form through a sequence of transformations. In a generation perspective, the features are translated to abstract morphemes which are then ordered, and expressed as concrete morphemes. The concrete templatic morphemes are interdigitated and affixes added, finally morphological and phonological rewrite rules are applied. In this section, we discuss our organization of linguistic knowledge, and give some examples; a more complete discussion of the organization of linguistic knowledge in MAGEAD can be found in (Habash et al., 2005).

[5] http://www.postgresql.org/

[6] http://wanna-b-a-bride.blogspot.com/2009/09/blog-post_29.html

[7] http://archive.hawaaworld.com/showthread.php?t=606067&page=76

[8] http://www.shabablek.com/vb/t40156.html

[9] http://forum.oujdacity.net/topic-t5743.html

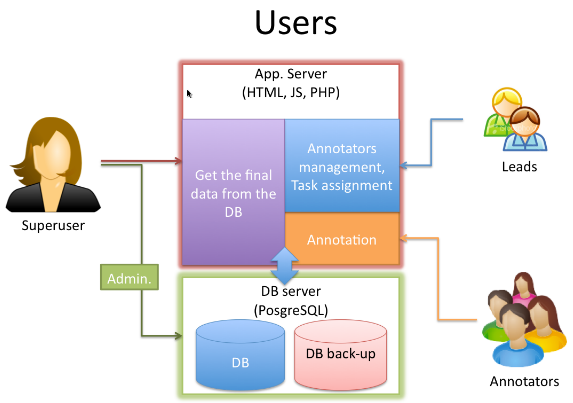

Figure 1: Servers and views organization.

Lexeme and Features Morphological analyses are represented in terms of a lexeme and features. We define the *lexeme* to be a triple consisting of a root, a *morphological behavior class* (MBC), and a meaning index. We do not deal with issues relating to word sense here and therefore do not further discuss the meaning index. It is through this view of the lexeme (which incorporates productive derivational morphology without making claims about semantic predictability) that we can have both a lexeme-based representation, and operate without a lexicon (as we may need to do when dealing with a dialect). In fact, because lexemes have internal structure, we can hypothesize lexemes on the fly without having to make wild guesses (we know the pattern, it is only the root that we are guessing). Our evaluation shows that this approach does not overgenerate. We use as our example the surface form أزدهرت *Aizdaharat* (*Azdhrt* without diacritics) "she/it flourished". The MAGEAD lexeme-and-features representation of this word form is as follows:

(1) Root:zhr MBC:verb-VIII POS:V PER:3 GEN:F NUM:SG ASPECT:PERF

Morphological Behavior Class An MBC maps sets of linguistic feature-value pairs to sets of abstract morphemes. For example, MBC verb-VIII maps the feature-value pair ASPECT:PERF to the abstract root morpheme [PAT_PV:VIII], which in MSA corresponds to the concrete root morpheme *V1tV2V3*, while the MBC verb-II maps ASPECT:PERF to the abstract root morpheme [PAT_PV:II], which in MSA corresponds to the concrete root morpheme *1V22V3*. We define MBCs using a hierarchical representation with non-monotonic inheritance. The hierarchy allows us to specify only once those feature-to-morpheme mappings for all MBCs which share them. For example, the root node of our MBC hierarchy is a word, and all Arabic words share certain mappings, such as that from the linguistic feature conj:w to the clitic *w+*. This means that all Arabic words can take a cliticized conjunction. Similarly, the object pronominal clitics are the same for all transitive verbs, no matter what their templatic pattern is. We have developed a specification language for expressing MBC hierarchies in a concise manner. Our hypothesis is that the MBC hierarchy is Arabic variant-independent, i.e. DA/MSA independent. Although as more Arabic variants are added, some modifications may be needed. Our current MBC hierarchy specification for both MSA and Levantine, which covers only the verbs, comprises 66 classes, of which 25 are abstract, i.e., only used for organizing the inheritance hierarchy and never instantiated in a lexeme.

MAGEAD Morphemes To keep the MBC hierarchy variant-independent, we have also chosen a variant-independent representation of the morphemes that the MBC hierarchy maps to. We refer to these morphemes as *abstract morphemes* (AMs). The AMs are then ordered into the surface order of the corresponding concrete morphemes. The ordering of AMs is specified in a variant-independent context-free grammar. At this point, our example (1) looks like this:

(2) [Root:zhr][PAT_PV:VIII]
[VOC_PV:VIII-act] + [SUBJSUF_PV:3FS]

Note that the root, pattern, and vocalism are not ordered with respect to each other, they are simply juxtaposed. The '+' sign indicates the ordering of affixival morphemes. Only now are the AMs translated to *concrete morphemes* (CMs), which are concatenated in the specified order. Our example becomes:

(3) <zhr,V1tV2V3,iaa> +at

Simple interdigitation of root, pattern and vocalism then yields the form *iztahar+at*.

MAGEAD Rules We have two types of rules. *Morphophonemic/phonological rules* map from the morphemic representation to the phonological and orthographic representations. For MSA, we have 69 rules of this type. *Orthographic rules* rewrite only the orthographic representation. These include, for example, rules for using the gemination *shadda* (consonant doubling diacritic). For Levantine, we have 53 such rules.

For our example, we get */izdaharat/* at the phonological level. Using standard MSA diacritized orthography, our example becomes *Aizdaharat* (in transliteration). Removing the diacritics turns this into the more familiar أزدهرت *Azdhrt*. Note that in analysis mode, we hypothesize all possible diacritics (a finite number, even in combination) and perform the analysis on the resulting multi-path automaton. We follow (Kiraz, 2000) in using a multi-tape representation. We extend the analysis of Kiraz by introducing a fifth tier. The five tiers are used as follows: Tier 1: pattern and affixational morphemes; Tier 2: root; Tier 3: vocalism; Tier 4: phonological representation; Tier 5: orthographic representation. In the generation direction, tiers 1 through 3 are always input tiers. Tier 4 is first an output tier, and subsequently an input tier. Tier 5 is always an output tier.

We implemented our multi-tape finite state automata as a layer on top of the AT&T two-tape finite state transducers (Mohri et al., 1998). We defined a specification language for the higher multi-tape level, the new MORPHTOOLS format. Specification in the MORPHTOOLS format of different types of information such as rules or context-free grammars for morpheme ordering are compiled to the appropriate LEXTOOLS format (an NLP-oriented extension of the AT&T toolkit for finite-state machines, (Sproat, 1995)). For reasons of space, we omit a further discussion of MORPHTOOLS. For details, see (Habash et al., 2005).

From MSA to Levantine and Egyptian We modified MAGEAD so that it accepts Levantine rather than MSA verbs. Our effort concentrated on the orthographic representation; to simplify our task, we used a diacritic-free orthography for Levantine developed at the Linguistic Data Consortium (Maamouri et al., 2006). Changes were done only to the representations of linguistic knowledge, not to the processing engine. We modified the MBC hierarchy, but only minor changes were needed. The AM ordering can be read off from examples in a fairly straightforward manner; the introduction of an indirect object AM, since it cliticizes to the verb in dialect, would, for example, require an extension to the ordering specification. The mapping from AMs to CMs, which is variant-specific, can be obtained easily from a linguistically trained (near-)native speaker or from a grammar handbook. Finally, the rules, which again can be variant-specific, require either a good morpho-phonological treatise for the dialect, a linguistically trained (near-)native speaker, or extensive access to an informant. In our case, the entire conversion from MSA to Levantine was performed by a native speaker linguist in about six hours. A similar but more limited effort was done to extend the Levantine system to Egyptian by introducing the Egyptian concrete morpheme for the future marker +هـ *h*+ ‘will’.

5. Resource Integration & Use: DIRA

DIRA (Dialectal Information Retrieval for Arabic) is a component in an information retrieval (IR) system for Arabic. It integrates the different resources created above in its pipeline. As mentioned before, one of the main problems of searching Arabic text is the diglossic nature of the Arabic speaking world. Though MSA is used in formal contexts on the Internet, e.g., in news reports, DA is dominant in user-generated data such as weblogs and web discussion forums. Furthermore, the fact that Arabic is a morphologically rich language only adds problems for IR systems. DIRA addresses both of these issues. DIRA is basically a query-term expansion module. It takes an MSA verb (and possibly some contextual material) as input and generates three types of surface forms for the search engine (the contextual material is left unchanged):

- **Mode 1:** MSA inflected forms. For example, the MSA query term أصبح *ÂSbH* ‘he became’ is expanded to several MSA forms including أصبحنا *ÂSbHnA* ‘we became’, سيصبح *sySbH* ‘he will become’, etc.
- **Mode 2:** MSA inflected with dialectal morphemes. It is common in DA to borrow an MSA verb and inflect it using dialectal morphology; we refer to this phenomenon as intra-word code switching. For example, the MSA query term أصبح *ÂSbH* can be expanded into هيصبح *hySbH* ‘he will become’ and هيصبحوا *hySbHwA* ‘they will become’.
- **Mode 3:** MSA lemma translated to a dialectal lemma, and then inflected with dialectal morphemes. For example, the MSA query term أصبح *ÂSbH* can be expanded into EGY بقى *bqý* ‘he became’ and هيبقى *hybqý* ‘he will become’.

Currently, DIRA handles EGY and LEV; with the existence of more resources for additional dialects, they will be added. The DIRA system architecture is shown in Figure 2. After submitting an MSA query to DIRA, the verb is extracted out of its context and sent to the MSA verb lemma detector, which is responsible for analyzing an MSA verb (using MAGEAD in the analysis direction) and computing its lemma (using MAGEAD in the generation direction). The next steps depend on the chosen dialects and modes. If translation to one or more dialects is required, the input lemma is translated to the dialects (Mode 3). Then, the MAGEAD analyzer is run on the lemma (MSA or DA, if translated) to determine the underlying morphemes (root and pattern), which are then used to generate all inflected forms using MAGEAD (again, which forms are generated depends on the mode). Finally, the generated forms are re-injected in the original query context (duplicates are removed).

6. Conclusions and Future Work

We presented COLABA, a large effort to create resources and processing tools for Dialectal Arabic. We briefly described the objectives of the project and the various types of resources and tools created under it. We plan to continue working on improving the resources and tools created so far and extending them to handle more dialects and more types of dialectal data. We are also considering branching into application areas other than IR that can benefit from the created resources, in particular, machine translation and language learning.

Acknowledgments

This work has been mostly funded by ACXIOM Corporation.

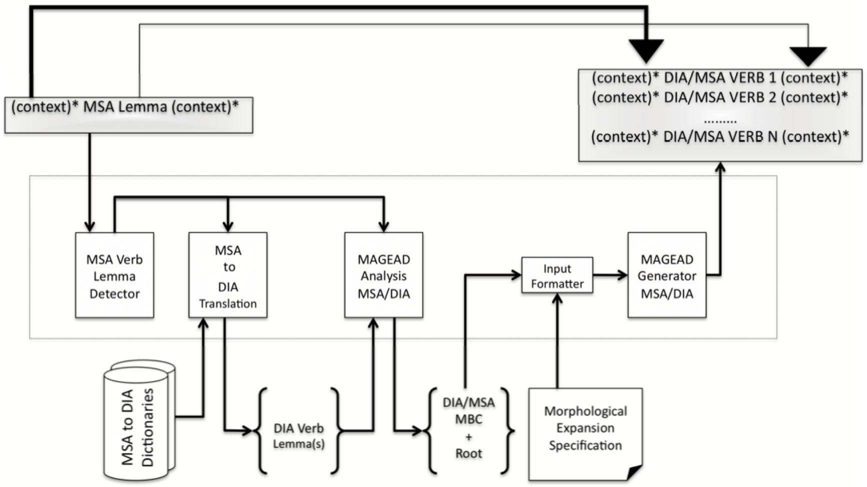
(context)* MSA Lemma (context)*
(context)* DIA/MSA VERB 1 (context)*
(context)* DIA/MSA VERB 2 (context)*
....
(context)* DIA/MSA VERB N (context)*
MSA Verb Lemma Detector
MSA to DIA Translation
MAGEAD Analysis MSA/DIA
Input Formatter
MAGEAD Generator MSA/DIA
MSA to DIA Dictionaries
DIA Verb Lemma(s)
DIA/MSA MBC + Root
Morphological Expansion Specification

Figure 2: DIRA system architecture

7. References

Ernest T. Abdel-Massih, Zaki N. Abdel-Malek, and El-Said M. Badawi. 1979. *A Reference Grammar of Egyptian Arabic*. Georgetown University Press.

Imad A. Al-Sughaiyer and Ibrahim A. Al-Kharashi. 2004. Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3):189–213.

Mohamed Altantawy, Nizar Habash, Owen Rambow, and Ibrahim Saleh. 2010. Morphological Analysis and Generation of Arabic Nouns: A Morphemic Functional Approach. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.

Mary Catherine Bateson. 1967. *Arabic Language Handbook*. Center for Applied Linguistics, Washington D.C., USA.

Yassine Benajiba and Mona Diab. 2010. A web application for dialectal arabic text annotation. In *Proceedings of the LREC Workshop for Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects*.

Kristen Brustad. 2000. *The Syntax of Spoken Arabic: A Comparative Study of Moroccan, Egyptian, Syrian, and Kuwaiti Dialects*. Georgetown University Press.

Tim Buckwalter. 2004. Buckwalter Arabic morphological analyzer version 2.0.

Mark W. Cowell. 1964. *A ReferenceGrammar of Syrian Arabic*. Georgetown University Press.

Wallace Erwin. 1963. *A Short Reference Grammar of Iraqi Arabic*. Georgetown University Press.

Nizar Habash and Owen Rambow. 2006. Magead: A morphological analyzer for Arabic and its dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Coling-ACL'06)*, Sydney, Australia.

Nizar Habash, Owen Rambow, and Geroge Kiraz. 2005. Morphological analysis and generation for Arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, MI.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

N. Habash, O. Rambow, M. Diab, and R. Kanjawi-Faraj. 2008. Guidelines for Annotation of Arabic Dialectness. In *Proceedings of the LREC Workshop on HLT & NLP within the Arabic world*.

Clive Holes. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.

H. Kilany, H. Gadalla, H. Arram, A. Yacoub, A. El-Habashi, and C. McLemore. 2002. Egyptian Colloquial Arabic Lexicon. LDC catalog number LDC99L22.

George Anton Kiraz. 2000. Multi-tiered nonlinear morphology using multi-tape finite automata: A case study on Syriac and Arabic. *Computational Linguistics*, 26(1):77–105.

Mohamed Maamouri, Tim Buckwalter, and Christopher Cieri. 2004. Dialectal Arabic Telephone Speech Corpus: Principles, Tool design, and Transcription Conventions. In *NEMLAR International Conference on Arabic Language Resources and Tools*.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. 2006. Developing and using a pilot dialectal Arabic treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC'06*, Genoa, Italy.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 1998. A rational design for a weighted finite-state transducer library. In D. Wood and S. Yu, editors, *Automata Implementation*, Lecture Notes in Computer Science 1436, pages 144–58. Springer.

Frank Rice and Majed Sa'id. 1979. *Eastern Arabic*. Georgetown University Press.

Richard Sproat. 1995. Lextools: Tools for finite-state linguistic analysis. Technical Report 11522-951108-10TM, Bell Laboratories.