# CODACT: Towards Identifying Orthographic Variants in Dialectal Arabic

**Pradeep Dasigi**
Computer Science Department
Columbia University
in the City of New York
pd2359@columbia.edu

**Mona Diab**
Center for Computational Learning Systems
Columbia University
in the City of New York
mdiab@ccls.columbia.edu

## Abstract

Dialectal Arabic (DA) is the spoken vernacular for over 300M people worldwide. DA is emerging as the form of Arabic written in online communication: chats, emails, blogs, etc. However, most existing NLP tools for Arabic are designed for processing Modern Standard Arabic, a variety that is more formal and scripted. Apart from the genre variation that is a hindrance for any language processing, even in English, DA has no orthographic standard, compared to MSA that has a standard orthography and script. Accordingly, a word may be written in many possible inconsistent spellings rendering the processing of DA very challenging. To solve this problem, such inconsistencies have to be normalized. This work is the first step towards addressing this problem, as we attempt to identify spelling variants in a given textual document. We present an unsupervised clustering approach that addresses the problem of identifying orthographic variants in DA. We employ different similarity measures that exploit string similarity and contextual semantic similarity. To our knowledge this is the first attempt at solving the problem for DA. Our approaches are tested on data in two dialects of Arabic - Egyptian and Levantine. Our system achieves the highest Entropy of 0.19 for Egyptian (corresponding to 68% cluster precision) and Levantine (corresponding to 64% cluster precision) respectively. This constitutes a significant reduction in entropy (from 0.47 for Egyptian and 0.51 for Levantine) and improvement in cluster precision (from 29% for both) from the baseline.

## 1 Introduction

Arabic is the native tongue of over 300M people world wide. The Arabic language exhibits a relatively unique linguistic phenomena known as diglossia (Ferguson, 1959) where two forms of the language live side by side: a standard formal form known as Modern Standard Arabic (MSA) and an informal spoken form, the vernaculars used in everyday communication referred to as Dialectal Arabic (DA). MSA is the only language of education and is used in formal settings and Broadcast news. The only written standard is in MSA using the Arabic script. Technically there are no native speakers of MSA. On the other hand, DA is the mother tongue for all native speakers of Arabic however it is not traditionally a written form of the language and it differs significantly enough from MSA on all levels of linguistic representation that results in huge inconsistencies in orthography. This was not a problem a decade ago from an NLP perspective since all the resources were in MSA. Now with the proliferation of online media and informal genres, DA is ubiquitous online. Users of DA online write in different scripts (Arabic, Romanizations interspersed with digits), they also sometimes write phonemically. Similar to other languages (not unique to DA) in these informal genres, we observe rampant speech effects such as elongations and the use of emoticons within the text which compounds the problem further for processing DA. If NLP tools want to process real Arabic as spoken by its people, they need to address DA seriously. This paper presents an initial attempt at addressing the pervasive inconsistencies in DA orthography in informal media.

We cast the problem of lack of DA orthographic standards as an identification of spelling variants problem using unsupervised clustering techniques. We evaluate our results against a gold corrected set of data in two dialects: Egyptian (EGY) and



*Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 318–326,
Chiang Mai, Thailand, November 8 – 13, 2011. 

Levantine (LEV). We focus our current efforts on identifying the orthographic variants in Arabic script though our work is extendible to the Romanizations as well. Such an identification is a necessary step for normalizing the variation which is useful for addressing the sparseness problem for DA. We contend that there are patterns in the variations that could be captured and processed. Also it is worth pointing out that this problem encompasses the spelling mistakes problem but it goes beyond it to address legitimate orthographic variants. Hence we attempt an approach that is generic enough to cover both scopes.

This paper is organized as follows: in Section 2 we show some of the variations between MSA and DA on different levels of linguistic representation; Section 3 discusses some related work; in section 4 we outline our approach and experimental conditions; in Section 5 we describe the data against which we evaluate our approach; we discuss the results and evaluation in Section 6; in Section 7 we discuss errors and performance of the system and approach proposed; finally, in Section 8 we conclude with some final remarks and a look at some future directions.

## 2 DA vs. MSA Phenomena

Most of the research effort, to date in creating tools and resources for Arabic has focused on MSA. In recent years we have seen a concentrated effort on making Arabic processing tools on par with English processing tools (Habash and Rambow., 2005; Diab et al., 2007; Kulick, 2010; Green and Manning., 2010). Researchers interested in handling realistic Arabic text have come to the realization that DA needs to be addressed. Applying state of the art MSA processing tools directly to DA yields very low performance proving the significant difference between the two varieties. For instance applying MSA tokenizers to DA yields a performance of 88% which is completely unacceptable as an initial processing tool performance. It is worth noting that state of the art MSA tokenization is at 99.2% (Diab et al., 2007). This low performance on DA can be explained by the genre differences (MSA tools are trained on newswire genres) but compared to English, we do not observe such a huge discrepancy between tokenizers trained on newswire when applied to informal genres. The significant drop in performance can be safely relegated to the inherent differences between the two varieties of Arabic. MSA differs from DA on the phonological, morphological, lexical, syntactic, semantic and pragmatic levels. The degree of variation depends on the specific dialect of Arabic. For instance, phonologically MSA would pronounce the word for dress as *fustAn* and spell it فستان while the same word in LEV is pronounced as *fusTAn*[1] with an emphatic T and could possibly be written phonemically in LEV as فوصطان. Morphologically, DA exhibits simpler inflectional morphology than MSA overall however cliticization is more nuanced rendering tokenization a more complex problem in DA than in MSA. For example, DA has lost all explicit marking of grammatical case and dual marking on verbal predicates. The MSA phrase *AlmwZfAn AklA*, الموظفان اكلا, meaning 'the employees_dual_nominative ate_dual', becomes *AlmwZfyn AlAtnyn AkAlw*, الموظفين الاتنين اكلو, 'the *two* employees_plural_no_case ate_plural'. Hence we note the loss of dual inflection marking and nominative case marking. On the other hand, clitization is more complex in DA as follows: EGY *mAHkytlhAlhw$*, ماحكيتلهالهوش, 'she did not recount it to him' is expressed in three words in MSA as *lm tHkyhA lh* لم تحكيها له. The lexical, syntactic, semantic and pragmatic variations abound between MSA and DA. The phonological and morphological differences lend themselves directly to the orthographic variation problem exhibited with DA. Writers of DA use a myriad of scripts to encode DA. All of which are used inconsistently even within the writings of the same author in the same post/article/blog. The most frequent scripts used are Arabic and Romanization. We note that people have use also Hebrew and Cyrillic scripts to write Arabic as well. For Arabic script we see inconsistencies in characters that exhibit regional variations such as the *qAf* sound ق. This letter is pronounced as a glottal stop ´ in EGY and LEV but as a *g* sound in the Gulf states and *q* sound in Tunisia, in most cases. Speakers and writers pertaining to these different dialects could render it in the orthography as it appears in a word such as *he said qAl* قال as *Ál* ال [EGY], *gAl* جال [Gulf] or *qAl* قال [Tunisian]. Moreover, we observe more severe variants in the Romanized script for the same word where the writers can render the EGY as *2Al*, *AAl*,

[1] We use the Buckwalter Arabic Transliteration standard for the Romanized Arabic throughout the paper. www.qamus.org

or *qAl*. For the purposes of this paper we will focus our discussion on identifying the orthographic variants only in the Arabic script leaving the handling of orthographic variants in Romanization for future work.[2]

## 3 Related Work

Most of the recent work in the area of orthographic variant detection and spelling correction has been towards resolving inconsistencies in spellings of Named Entities(NE). Huang et al. (2008) use NE spelling variant detection to improve the performance of Machine Translation (MT) and Information Extraction (IE) systems. (Habash and Metsky, 2008) cluster Urdu phrases mapping to the same English phrase to automatically learn morphological variation rules. Since Urdu is a morphologically rich language, such variations result in many OOV words. They use the these rules learned, as a part of MT system to replace OOV Urdu words with in-vocabulary words online. Raghavan and Allan (2005) use the edit distance metric along with generative models trained from Automatic Speech Recognition (ASR) output to cluster queries to improve performance of Information Retrieval (IR) systems. Bhagat and Hovy (2007) attempted to generate all possible spelling variations of a person's name. One method is supervised and uses CMU speech dictionary to train a phonetic model. Another is to cluster a large set of names that are known to sound similar using Soundex (Knuth, 1973). Although some earlier work related to spelling variations (Golding and Roth, 1999) dealt with the generalized problem, most of the recent work is confined to NEs.This is because of the relevance of the problem to NLP applications such as MT, IE, IR and ASR. Accordingly we note that the problem we try to solve is more generic since the lack of orthographic standard in DA affects the spellings of all kinds of words.

## 4 Approach

Our goal is to identify orthographic variations in textual DA. We build a system, CODACT, that aims at identifying and eventually normalizing such DA orthographic variants. We use techniques noted in the spelling correction literature. Our approach is mainly unsupervised. We view the problem as a clustering problem where our goal is to identify if two strings are similar, and hence cluster together. To that end we explore three basic similarity measures: (a) String based Similarity as direct Levenshtein Edit Distance; (b) String based Similarity Biased Edit Distance; and ( c) Contextual String Similarity. We model the strings of interest in a vector space. We build a matrix for the string types of interest. We induce the clusters from the matrix by grouping the strings in the row entries together based on the similarity of their respective vectors in the matrix. We use Cosine Similarity between vectors and we use the implementation of the CLUTO Repeated Bisection (RB) algorithm with cosine similarity being the measure of similarity between vectors. (Zhao and Karypis, 2001). CLUTO is very suitable for clustering high dimensional datasets. CLUTO's repeated bisection partition method is used for clustering. In this method, for obtaining a *k* -way clustering, *k-1* repeated bisections are made. Each partition is made to the input dataset such that the clustering criterion function is optimized.

The row entries for the matrix are referred to as the focal string types of interest. We vary the dimensions as follows: (a) for N focal words, we have the same N focal words in the matrix dimensions, yielding an NxN matrix; or (b) the dimensions are all the string types in the corpus of interest yielding an NxM matrix. The cells of the matrix are populated based on one of the different similarity measures or a combination of them after normalization. We describe the different similarity measures next.

### 4.1 String Based Similarity Metrics

Strings that vary from each other minimally are likely to be orthographic variants of one another. Following this intuition, strings are grouped based on their string edit distance. We explore the basic known Levenshtein Edit Distance measure as in (Levenshtein, 1966) (LEDM). Moreover we extend the LEDM to account for known phonological variations on the character level. We refer to this as the Biased Edit Distance Metric (BEDM). BEDM has the same exact formulation as LEDM as a metric however it is more relaxed in that it treats letters that are considered similar as if they are the same, i.e. they are not *substitutions* of each

[2]It is worth noting that tools such as Yamli and Maren which transliterate Romanization to Arabic script serve as in interesting platform for handling the Romanization problem that could be easily leveraged.

other. The intuition behind adding such a bias is the fact that Arabic letters may have different pronunciations depending on the context. For example, the letter ا might have a sound equivalent to the any of letters أ, آ, و and ي. This is ignored by LEDM, and they are treated as different letters therefore incurring the substitution penalty. When BEDM is applied, any two letters that have the same sound are treated as a match. For example, (1) *fstAn*, 'dress', فستان, and the possible variants (2) *fstAn*, (3) *fSTAn*, (4) *fsTAn*, and (5) *fjTAn* would have the following calculations: (1) and (2) would be a perfect match, i.e. a distance of 0 according to both LEDM and BEDM; (1) and (3) would be penalized for substituting *S, T* for *s,t* in (1), therefore a distance of 0.4 according to LEDM, however for BEDM *s* and *S* are considered similar to each other and so are *t* and *T*, then the distance of (1) and (3) is 0. Similarily for (1) and (4) according to LEDM the distance is 0.2, but for BEDM the distance is 0. For (1) and (5) the LEDM will be 0.4 and for BEDM it will be 0.2. Hence, BEDM is a more nuanced and relaxed form of LEDM. The list of similar letters is taken from scholar seeded studies of phonological variations across different DA. The list is rendered in Table 1. We refer to this list as sound change rules (SCR).[3] The SCR are not always symmetric, for example a *v* can be replaced with a *S* but not vice versa.

### 4.2 Contextual String Similarity

We explore another relatedness measure of contextual string similarity (CSS). The intuition is that if two strings are variants of each other as they are semantically similar, they are bound to appear with similar contexts. Accordingly we model this notion via representing strings with their context co-occurrence vectors. In this framework , we represent the co-occurrence frequency of the focal string and dimensional string in all the sentences in the corpus within a window of 3 tokens. The observations are aggregated and used in the cell. If the focal string and the dimensional string never co-occur, then the cell value is set to 0. Contextual Similarity Metric (CSS) between two words is defined as a cosine similarity between their context vectors.

[3]We are aware that this list can be further refined to reflect the specific dialect under investigation. We plan to incorporate a better customized SCR depending on the variety of DA.

| Letter | Similar Sounding Letters |
|---|---|
| A | {, <, >, ', &, }, $w$, $y$, \| |
| ' | A,{, }, <, >, \|, $y$, &,w |
| } | A, y, &, ', {, <, >, \|,w |
| & | A, y, }, ', {, <, >, \|,w |
| \| | A, y, ', {, <, >, } |
| { | A, y, &, ', }, <, >, \| |
| < | A, ', {, >, \|, } |
| > | A, ', {, <, \|, } |
| t | T, v |
| v | s, t, S |
| j | q, y, $ |
| H | h, E |
| d | $*$, D |
| $*$ | d, z, Z |
| z | $*$, Z, d |
| s | $, S, v |
| $ | s, v |
| S | s |
| D | Z, d, z, $*$ |
| T | S, Z, t |
| Z | T, D, z, d, $*$ |
| E | H |
| g | E, x |
| q | ', A, }, k, j |
| k | q |
| h | p, A |
| p | h, t |
| w | &, A, Y |
| y | }, A, Y |
| Y | y, A |

Table 1: Sound Change Rules for Arabic Letters as obtained from Linguistic Studies

### 4.3 Experimental Conditions

We experimented with each of these measures in isolation and in combination. In the case of combination, we normalized the values of metrics. Table 2 illustrates the values contained in the cells of the constructed matrices in the different conditions.

We have two different matrix dimension sizes depending on how extensive the feature space is. The first case is NxN, meaning that the set of words corresponding to both the rows and columns of the matrix are the focal words. The second case has all the unique words in the corpus representing the columns, making it NxM. This yields 6 isolated conditions and 8 combined conditions.

| Metric | Cell Values |
|---|---|
| LEDM | 1.0 - Normalized Levenshtein Distance |
| BEDM | 1.0 - Normalized Levenshtein Distance biased by phonetic similarity across letters |
| CSS | Co-occurrence frequencies |
| LEDM+BEDM | Mean of LEDM and BEDM |
| CSS+LEDM | Mean of LEDM and Normalized CSS (both are in range $\{0, 1\}$) |
| CSS+BEDM | Mean of BEDM and Normalized CSS |
| CSS+LEDM+BEDM | Mean of LEDM, BEDM and Normalized CSS |

Table 2: Matrix Cell Values

- LEDM-NxN where the similarity measure is a LEDM and the matrix size is NxN
- BEDM-NxN where the similarity measure is a BEDM and the matrix size is NxN
- CSS-NxN where the similarity measure is a CSS and the matrix size is NxN
- LEDM-NxM where the similarity measure is a LEDM and the matrix size is NxM
- BEDM-NxM where the similarity measure is a BEDM and the matrix size is NxM
- CSS-NxM where the similarity measure is a CSS and the matrix size is NxM
- LEDM-BEDM-NxN where the similarity measure is a combination of LEDM and BEDM and the matrix size is NxN
- LEDM-CSS-NxN where the similarity measure is a combination of LEDM and CSS and the matrix size is NxN
- BEDM-CSS-NxN where the similarity measure is a combination of CSS and BEDM and the matrix size is NxN
- LEDM-BEDM-CSS-NxN where the similarity measure is a combination of LEDM, BEDM, and CSS the matrix size is NxN
- LEDM-BEDM-NxM where the similarity measure is a combination of LEDM and BEDM and the matrix size is NxM
- LEDM-CSS-NxM where the similarity measure is a combination of LEDM and CSS and the matrix size is NxM
- BEDM-CSS-NxM where the similarity measure is a combination of CSS and BEDM and the matrix size is NxM
- LEDM-BEDM-CSS-NxM where the similarity measure is a combination of LEDM, BEDM, and CSS the matrix size is NxM

## 5 Evaluation Data

In order to measure the performance of this approach we need data that has the variants identified. We created such data by asking native speakers of DA to normalize variants into a standard conventionalized form in Arabic script. We targeted two dialects of Arabic: Egyptian (EGY) and Levantine (LEV). For EGY we specifically focused on Cairene Egyptian. For Levantine, we had a collection of Palestinian, Jordanian, Lebanese and Syrian Dialectal data. Most of the data is considered Syrian in our LEV collection however. Both data sets for both DA are derived from the web (Diab et al., 2010). The data is part of a larger collection we refer to in this paper as COMMENTDA collection. COMMENTDA comprises 3M token strings for EGY and 3M token strings for LEV. For EGY we had 2 annotators and an adjudicator, and for LEV we had 4 annotators and an adjudicator. The annotators were instructed to identify tokens that are considered incorrect orthographically according to a specific convention that we devised known as CODA (Conventionalized Orthography for Dialectal Arabic) after being trained on CODA (Habash et al., 2011). The annotators we asked to identify three different classes of variation from the CODA convention: (a) change in spelling of a string which included dealing with speech effects such as elongations, (b) introduction of spaces or splitting a string into multiple strings, and (c ) deletion of spaces or merging strings. Many corrections included simultaneously both a spelling change and a split or merge of a string as well. Table 3 gives detailed statistics of the annotated data for EGY and LEV, respectively.

| Category | EGY | | LEV | |
|---|---|---|---|---|
| | **Tokens** | **Types** | **Tokens** | **Types** |
| Inspected | 39328 | 12703 | 74450 | 19045 |
| Spelling Changes Only | 6835 | 3651 | 5877 | 3519 |
| Splits | 1373 | 751 | 1013 | 590 |
| Splits Only | 752 | 288 | 768 | 169 |
| Merges | 633 | 440 | 789 | 598 |
| Two String Merges | 72 | 61 | 119 | 109 |
| More Than Two String Merges | 561 | 379 | 670 | 489 |
| Merges Only | 374 | 211 | 357 | 264 |
| Unique Changes | 7961 | 4150 | 7002 | 3952 |
| All Changes Including Overlaps | 9248 | 4720 | 7913 | 4326 |
| Unchanged | 30080 | 7983 | 66537 | 14719 |
| Common Strings Across Annotators | 1541 | 843 | 3271 | 1759 |
| Inter-Annotator Agreements | 1080 (70.08%) | 576 (68.33%) | 2688 (82.18%) | 1489 (84.65%) |

Table 3: Annotation Statistics

It is worth noting that roughly 24% of the EGY data had changes of different types on the token level corresponding to 37% of changes to the types for EGY. For LEV, only 11% of the tokens were changed corresponding to 23% of the types that were changed. This suggests that the EGY data had a lot more variability. It was actually noted that a lot of the EGY data was not consistently EGY but rather from other DA compared to LEV that was considered relatively homogeneous. The last row in the table shows the number of cases where annotators agreed with each other on the correction. It also shows their percentages to the number of common annotations as shown by the row above it.

We created the gold data clusters of variants by grouping all the strings that are mapped to the same corrected CODA form. This data consisted of 290 clusters of strings in LEV with an average of 2.6 orthographic variants per cluster, and 312 string clusters in EGY with an average of 2.5 variant per cluster. All our experiments are conducted on surface forms of the strings with no preprocessing.

## 6 Evaluation

In order to derive statistics to build our matrices, we use two data sets: COMMENTDA and an augmented data set (RCorpora) which is double the size of COMMENTDA for each dialect. RCorpora comprises 6M strings for EGY, and 6M strings for LEV. In the NxN matrix conditions, the size of the corpora used to derive the statistics only affects the conditions involving CSS. In the NxM conditions, the corpora sizes affect the number of matrix dimensions as well as the cell values for the CSS conditions. Application of seven metrics to four class-data size combinations gives 25 distinct runs per dialect since in NxN case, augmenting the data does not change the metrics LEDM, BEDM, and LEDM+BEDM. It has to be noted that for NxM experiments this is not the case. Table 4 gives the various statistics on the different data sizes of unique string types.

Each clustering output is compared with the gold-standard clusters using Purity and Entropy measures (Zhao and Karypis, 2001). Every word in a given cluster in the output belongs to one or more gold clusters. These gold clusters are referred to as relevant gold clusters of the given output cluster. Purity or precision of a cluster is the fraction of its words in its relevant gold clusters. Entropy gives the measure of ambiguity in the clustering output. The larger the number of word in a relevant gold cluster, the higher the entropy value. In addition to these measures, the value of recall is also calculated. This equals the fraction of words in the relevant gold clusters that are in the given cluster. Together these three measures give a complete assessment of the quality of the output clusters. As a baseline for comparison, where strings are randomly assigned to clusters assuming the gold number of clusters per dialect. Table 5 shows the results of all the experiments described.

| | COMMENTDA | | COMMENTDA+RCorpora (C+R) | |
|---|---|---|---|---|
| | **EGY** | **LEV** | **EGY** | **LEV** |
| NxN | 729 | 717 | 729 | 717 |
| NxM | 205088 | 237598 | 433697 | 410206 |

Table 4: Data Size Variations in the two dimensions conditions

## 7 Discussion

All the systems outperform the random baseline. The best results are presented in bold in Table 5.

**Phonological Bias** From the results, it can be seen that of the individual metrics, BEDM consistently performs better than LEDM. This shows that introducing a phonological bias while matching letters does have a significant positive effect in identifying spelling variants. We believe that this effect will be more pronounced if the SCR are more tailored to the specific dialect under study. We note that CSS has the worst performance of the three individual metrics in all cases. This may be attributed to the level of processing of the data. The data is dealt with specifically on the surface level and Arabic being a very rich morphological language results in a very sparse distribution of forms. This can be mitigated by using even larger corpora. Typically in such studies that rely on distributional similarity an order of magnitude larger than what we employed is exploited. We relegate this to future work.

**Combined Metrics** Combination of LEDM and BEDM showed better precision, recall, and entropy than each of them in isolation in every case for both dialects. Although adding CSS has shows improvement in LEDM, mainly in EGY, it actually worsened the performance of BEDM. This is more evident when CSS+LEDM+BEDM is compared with LEDM+BEDM. Almost all the quality measures show that CSS metric adds noise.

**NxM vs. NxN** Using a bigger class of words as a feature set for vector similarity significantly improves the performance of the system. This holds for all the metrics in both the dialects.

**Data Augmentation** Although there are slight improvements due to increase in data, on the whole this does not seem to affect the performance of the system significantly. We only see some effect in the CSS measures which is expected.

### 7.1 Error analysis

| Cluster type | Members |
|---|---|
| Gold | <n$' w<n$' An$A' |
| LEDM | w<n$ w<n$' wgnY wgnYY |
| BEDM | <n$' An$A' w<n$' |
| LEDM+BEDM | <n$' An$A' w<n$' |

The example above shows a comparison between LEDM and BEDM metrics. The clusters in second and third columns are the ones nearest to the gold cluster in first column. It can be observed that An$A' is not a part of the LEDM cluster since the word has more dissimilar letters. However, BEDM captures the fact that the words are phonologically similar. The example below shows a similar pattern too. LEDM differentiates ||mdh from the other words too much to identify it to be a potential variant.

| Cluster type | Members |
|---|---|
| Gold | jAmddp ||mdh jAAmdp jAmdh |
| LEDM | jAmddp jAAmdp jAmdh |
| BEDM | jAmddp ||mdh jAAmdp jAmdh |
| LEDM+BEDM | jAmddp ||mdh jAAmdp jAmdh |

The next example illustrates the advantage of combining metrics. BEDM gives both higher precision and recall than LEDM when compared to the gold cluster. However, both of them do not have perfect precision or recall while the combined metric has both.

| Cluster type | Members |
|---|---|
| Gold | btAEtY btEty |
| LEDM | btAEtY bnt bt |
| BEDM | btAEtY btEty ty |
| LEDM + BEDM | btAEtY btEty |

A relaxed similarity metric does not necessarily result in higher clustering recall. In the next example, BEDM gives a lower recall than LEDM. However, the combined metric does better than the individual edit distance metrics in this example too.

| Cluster type | Members |
|---|---|
| Gold | byqwlw byqwlh byqwlwA byqlh |
| LEDM | byqwlw byqwlh byqwlwA yqwlp yqwlh |
| BEDM | byqwlw byqwlwA |
| LEDM + BEDM | byqlh byqwlh byqwlw byqwlwA |

| Dialect | Matrix Size | Similarity Metric | Quality Metric | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Precision | | Recall | | Entropy | |
| | | | C | C+R | C | C+R | C | C+R |
| | | Baseline | 29.63 | | 34.68 | | 0.4728 | |
| | | LEDM | 44.37 | | 50.33 | | 0.3185 | |
| | | BEDM | 58.85 | | 63.9 | | 0.2342 | |
| | | LEDM+BEDM | 61.74 | | 67.22 | | 0.2084 | |
| | **NxN** | CSS | 25.21 | 23.45 | 31.12 | 34.2 | 0.2779 | 0.2575 |
| | | CSS+LEDM | 45.03 | 44.59 | 50.05 | 49.86 | 0.3184 | 0.3207 |
| | | CSS+BEDM | 55.83 | 56.29 | 63.8 | 63.04 | 0.2433 | 0.238 |
| **EGY** | | CSS+LEDM+BEDM | 61.91 | 59.71 | 66.46 | 66.93 | 0.2218 | 0.2229 |
| | | LEDM | 53.96 | 55.07 | 56.5 | 57.45 | 0.3047 | 0.3079 |
| | | BEDM | 61.41 | 59.04 | 65.23 | 63.33 | 0.2401 | 0.2425 |
| | | LEDM+BEDM | **67.14** | **68.51** | **69.97** | **71.2** | **0.2006** | **0.196** |
| | **NxM** | CSS | 35.94 | 36.06 | 36.57 | 38.09 | 0.4184 | 0.4074 |
| | | CSS+LEDM | 54.54 | 56.71 | 56.69 | 58.3 | 0.3027 | 0.2981 |
| | | CSS+BEDM | 60.63 | 58.79 | 64.28 | 63.9 | 0.2419 | 0.2393 |
| | | CSS+LEDM+BEDM | 66.26 | 68.23 | 69.69 | 70.92 | 0.1997 | 0.1988 |
| | | Baseline | 28.67 | | 31.10 | | 0.5177 | |
| | | LEDM | 51.44 | | 54.66 | | 0.2831 | |
| | | BEDM | 54.22 | | 60.61 | | 0.2403 | |
| | | LEDM+BEDM | 63.92 | | 64.67 | | 0.2226 | |
| | **NxN** | CSS | 30.4 | 23.4 | 24.11 | 30.07 | 0.2142 | 0.2602 |
| | | CSS+LEDM | 50.82 | 51.18 | 54.49 | 55 | 0.2883 | 0.2808 |
| | | CSS+BEDM | 54.62 | 56.12 | 62.51 | 60.7 | 0.2338 | 0.2459 |
| **LEV** | | CSS+LEDM+BEDM | 60.6 | 61.41 | 64.49 | 64.41 | 0.2156 | 0.2259 |
| | | LEDM | 57.11 | 56.76 | 57.68 | 57.38 | 0.2728 | 0.2797 |
| | | BEDM | 61.54 | 64.18 | 64.06 | 65.31 | 0.2305 | 0.2132 |
| | | LEDM+BEDM | **65.37** | **64.98** | **65.79** | **65.23** | **0.2091** | **0.205** |
| | **NxM** | CSS | 45.21 | 37.53 | 27.48 | 34.21 | 0.3148 | 0.3992 |
| | | CSS+LEDM | 57.61 | 55.36 | 57.33 | 57.2 | 0.2731 | 0.2745 |
| | | CSS+BEDM | 59.15 | 63.17 | 62.6 | 65.06 | 0.2234 | 0.2113 |
| | | CSS+LEDM+BEDM | **64.17** | **64.34** | **65.36** | **65.92** | **0.2099** | **0.1998** |

Table 5: Results

## 8 Conclusion and Future Work

We compared surface and contextual metrics for identifying spelling variants in DA. We also evaluated all the combinations of those metrics. We present an initial system CODACT. Our results hold clear cross-dialectal trends, showing that string similarity metric with a phonological bias, combined with simple edit distance as a similarity metric is better for this task than raw contextual similarity when the data is limited. The next step in this approach is to refine the co-occurrence model used in our approach. Using lemma forms instead of the surface forms can yield a potential improvement since Arabic is a morphologically rich language. Eventually we plan to develop a system that will automatically normalize orthographic variations in Dialectal Arabic to the CODA convention.

## References


[Golding and Roth1999] Andrew R. Golding and Dan Roth. 1999. *A winnow-based approach to context-sensitive spelling correction* Machine Learning, 34(1-3):107–130.

[Ferguson1959] Charles A. Ferguson. 1959. *Diglossia, Word*. Journal of the Linguistic Circle of New York, Vol. 15, No. 2, August 1959, pp. 325-340

[Habash and Metsky2008] Nizar Habash and Hayden Metsky 1959. *Automatic Learning of Morphological Variations for Handling Out-of-Vocabulary Terms in Urdu-English Machine Translation*. 8th AMTA conference, Hawaii, 21-25 October 2008

[Knuth1973] Donald E. Knuth. *The Art of Computer Programming – Volume 3: Sorting and Searching* Addison- Wesley Publishing Company, 1973.

[Huang et al.2008] Fei Huang , Ahmad Emami and Imed Zitouni. *When Harry Met Harri: Cross-lingual Name Spelling Normalization.* Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 391–399.

[Hema Raghavan and James Allan2005] Hema Raghavan and James Allan. *Matching Inconsistently Spelled Names in Automatic Speech Recognizer*

*Output for Information Retrieval.* Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, 2005.

[Kernighan et al.1990] Mark D. Kernighan, Kenneth W. Church, and William A. Gale. 1990. *A spelling correction program based on a noisy channel model*. Proceedings of COLING-90, pages 205–210

[Diab et al.2010] Mona Diab, Nizar Habash, Owen Rambow, Mohamed Al Tantawy, Yassine Benajiba. 2010. *COLABA: Arabic Dialect Annotation and Processing.*. Proceedings of the Workshop on Semitic Language Processing, LREC, May, Malta

[Habash et al.2011] Nizar Habash, Mona Diab, Owen Rambow. 2011. *Conventional Orthography for Dialectal Arabic (CODA) V.1.0.*. Technical Report 137382, `http://academiccommons.columbia.edu/catalog/ac:137382`, Columbia University, New York, NY, USA

[Diab et al.2007] Mona Diab, Kadri Hacioglu and Daniel Jurafsky. 2007. *Automated Methods for Processing Arabic Text: From Tokenization to Base Phrase Chunking*. In Arabic Computational Morphology: Knowledge-based and Empirical Methods.

[Habash and Rambow.2005] Nizar Habash and Owen Rambow. 2005. *Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop*. Proceedings of the Conference of American Association for Computational Linguistics (ACL'05).

[Bhagat and Hovy.2007] Rahul Bhagat and Eduard Hovy. 2007. *Phonetic Models for Generating Spelling Variants*. Proceedings of the 20th international joint conference on Artifical intelligence.

[Green and Manning.2010] Spence Green and Christopher D. Manning. 2010. *Better Arabic Parsing: Baselines, Evaluations, and Analysis*. Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010).

[Kulick2010] Seth Kulick. 2010. *Simultaneous Tokenization and Part-of-Speech Tagging for Arabic without a Morphological Analyzer*. Proceedings of the ACL 2010 Conference Short Papers.

[Levenshtein1966] V. I. Levenshtein. 1966. *Binary codes capable of correcing deletions,insertions and reversals*. Soviet Physics Doklady, Vol. 10, p.707–710.

[Zhao and Karypis2001] Ying Zhao and George Karypis. 2004. *Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering*. Machine Learning, Volume 55 Issue 3, June 2004.