# Automatic Verb Detection in Libyan Dialect

Abdusalam F Ahmad Nwesri[1] and Nabila Almabrouk S. Shinber[2]

[1] Faculty of Information Technology, University of Tripoli, Tripoli, Libya
[2] College of Science and Technology, Tripoli, Libya

[1]a.nwesri@uot.edu.ly, [2]shinbir@tcst.edu.ly

**Abstract.** Automatic recognition of verbs is crucial to a wide range of natural language processing tasks. Verbs exhibit the relational information in a sentence between the action and its participant and are considered the primary source of information in understanding a sentence and the base for any NLP task. In this paper, we experiment six machine learning algorithms to identify verbs from other words in the Libyan dialect. Among algorithms used, the Support vector classifier (SVC) was best at identifying verbs with a micro F1 score of 70%.

**Keywords:** Libyan dialect; Verbs; NLP; Machine Learning

## 1 Introduction

Libyan dialect (LD) is a modified version of classical Arabic. It is spoken in Libya with more than three main sub-dialects scattered in the east, west and the south of Libya. The dialect exhibits distinctive morphological features that differentiate it from Modern Standard Arabic (MSA) and other Arabic dialects. These features reflect historical influences, linguistic changes, and dialectal variations within Libya itself.
Although the MSA is still the main language used in formal written communications, the LD started to emerge as an informal written communication language within the social media and the Internet. Ordinary people find it easy to use in their writing than using MSA. With this change, the amount of Libyan dialect text in the Internet has increased and yet there are no proper NLP techniques that can effectively process the dialect.

The motivation behind this study is twofold. It is first a step to a larger project from which we aim to build a lexicon for LD, which does not yet exist. Second, most of machine learning techniques work more appropriately with MSA because they were trained mainly with data gathered from MSA sources. The problem, however, is that MSA are used in different genres which differ lexically and stylistically [1]. Our goal, therefore, is to test different machine learning approaches on LD text. In the next section we will present LD and how it differs from MSA, justify the need to have new NLP techniques to deal with it. In the following sections, we present the related work and the details of the experiments we carried out and their results.

### 1.1 Difference Between MSA and LD

MSA and LD are both considered as different versions of standard Arabic. MSA is used across the Arabic world in formal communications while LD is used in Libya for informal spoken communications. Nowadays LD started to emerge as a written form in social media, emails, blogs and SMS. Although there has been a considerable effort on studying MSA, most of these studies cannot be applied directly to LD due to the fact that LD differs from MSA in many aspects including morphology, phonology, syntax and Lexicon [2].

There are several morphological differences between LD and MSA. For example, the verb conjugation system in LD displays certain deviations from MSA patterns. It employs unique verb forms and conjugations specific to the dialect. For example, the use of the prefix "ن /n-/" for the 1st person in the present tense conjugation, such as "نكتب /niktib/" (I write) instead of the prefix "أ /a/" in MSA "أكتب /aktub/ " (I write). Another example is the replacement of dual suffixes used in MSA with those used for plurals. An instance instead of writing "نابتكى /yaktuban/", LD speakers would say "وبتكى /yiktbu/". In general, the verb inflectional complexity in LD is morphologically complex as a concatenative stem-based system [2].

The declension of nouns in LD may differ from MSA. It often simplifies the case system, with fewer distinct cases and more reliance on prepositions to indicate relationships. This simplification is particularly noticeable in the spoken language, where the accusative case is frequently omitted. Plural forms in LD can deviate from MSA patterns. While some plural forms align with standard Arabic plurals, others exhibit unique pluralization strategies. For instance, the use of the suffix "سى /-ees/" in the plural form, such as "سيطاطق /qitatees/" (cats) instead of the MSA "طط /qitat/" (cats).

Like any living language, the LD incorporates loanwords from various sources, such as other Arabic dialects, Italian, Turkish, and Berber languages. These loanwords contribute to the lexical richness of the dialect and reflect historical interactions with different cultures and languages.

Additionally the word order in LD as in other Arabic dialects is usually SubjectVerb-Object as in "دمحم بتك سردلا /Mohamed ktab addars/" (Mohamed wrote the lesson), while in MSA it is Verb-Subject-Object as in "كتب دمحم الدرس /kataba Mohamed addarsa/" [3].

The difference between LD and MSA enforces the need to test NLP techniques used on MSA and other dialects to LD. In this study our main concern is to identify verbs in LD in order to lay down the basement for further NLP research.

### 1.2 Verbs in Arabic

Verbs have long posed a challenge to automatic natural language understanding. the task of acquiring verb-related information from corpora is seen as an important step toward better machine understanding of text [4].

Verbs are considered the basic building block of Arabic words, as most of Arabic words are formed from the three-letter roots of past tense verbs. They are central to the meaning of the sentence.

Recognizing verbs in Arabic text faces several challenges. First verbs are not always written as separate words as in English. However, they usually come attached with prefixes and suffixes. In some cases, the sentence contains the verb only with no other words except affixes. for example: the token "اهنوبتكيس /sayaktubunaha/" meaning "they will write it" is a complete sentence that contains the past tense verb "كتب /kataba/ wrote" with two prefixes and two suffixes. Similarly the word "اهوبتكيب /ebyktbuha/" is the equivalent word in LD with two prefixes "ب " "/will/" and "ي " "/present verb prefix/" and two suffixes "-و" masculine plural suffix and "اهـ" singular feminine suffix.

Lack of diacritics in written Arabic text adds extra layer of difficulty in recognizing verbs from nouns for example, the word "رعش " is ambiguous when it comes alone without diacritics. it can be read as "شَعَرَ /sh'ara/ felt (v.)", "شَعْرٌ /sh'aron/ hair (n.)", or "شِعْرٌ /shi'ron/ poetry (n.)". The same word in LD carries the same meaning although pronounced differently.

Early research on recognizing different parts of speech including verbs relied on manual or rule-based lexicons [5, 6], however, with the advancement of machine learning, it is now possible to identify verbs and learn their semantic and syntactic meanings based on their statistical existence in text corpora. Most recent studies on morphological analysis of verbs and other speech parts use machine learning (ML). different algorithms used to categorize parts of speech in Arabic text [7] as well as dialectal text [3].

Our aim in this study is to test the automatic detection of LD verbs using machine learning algorithms. In this paper we test six different machine learning algorithms namely: Logistic Regression (LG), Decision Trees (DT), Na¨ıve Bayes (NB), Neighborhood Classifier (NC), Support Vector Classifier (SVC), and Random Forests (RF) on identifying verbs in the LD.

## 2 Related Work

Early studies on analyzing Arabic text focused on developing Part of Speech (PoS) taggers [5, 6, 8, 9]. These studies focused on developing a tag-set for Arabic language that can be used to label different parts of speech inside the text. Morphological analysis was used in automatic tagging. In the last decade, automatic PoS tagging has been extended to include Arabic dialects mostly using artificial intelligence [7, 8, 10, 11].

There are few studies focused on verb identification alone. Technology used to find verbs in Arabic text varies between these studies. Othman et al. [12] used regular expressions morphological model to identify verb patterns in Arabic text. Their approach detected 87% of verbs in the first four Surat of the holy Quran. Azman [13] built an Arabic model to find root verbs from surface words. Surfaces forms of verb are structured in tree hierarchy, putting the root verb as the tree's root and followed with some levels which represent different surface forms of the verb. Their system"RootIT" is claimed to achieve 97.34% on F1 measure in identifying correct root verb.

A recent work carried by Ahmed and Tosun [14] shows that finding Arabic verbs using roots and patterns without affix removal. They used 17 patterns to represent verbs in Arabic. They claimed that their approach is reliable, however, there was no proper evaluation in their work.

With the advent of new technologies such as ML which needs less human intervention, interest in studying Arabic dialects increased in the past decade. Alharbi et al. [3] introduced a PoS tagger for the Gulf dialect. They used two machine learning methods, namely Support Vector Machine (SVM) classifier and bi-directional Long Short Term Memory (Bi-LSTM) for sequence modeling. Their methods achieved 91% in accuracy using the Bi-LSTM technique. Darwish et al. [11] introduced a multi-dialect Arabic PoS tagging approach. They used Conditional Random Fields (CRF) sequence labeler to train POS taggers on Egyptian, Levantine, Gulf, and Maghrebi tweets. Their joint model was able to tag the four dialects with an average accuracy of 89.3%. This work has been extended to use deep neural network with stacked layers of convolutional and recurrent networks with the CRF output layer. They achieved 92.4% accuracy across all four dialects [15].

## 3 Experiments

In this section, we present our experiments. we first describe our datasets, then we run six different machine learning algorithms namely: Logistic Regression (LG), Decision Trees (DT), Na¨ıve Bayes (NB), Neighborhood Classifier (NC), Support Vector Classifier (SVC), and Random Forests (RF) on these datasets to assess their ability to identify verbs from other words.

### 3.1 Datasets

Two datasets have been used, the first was collected from articles and LD stories published on social media. The dataset contained 4712 unique words which are manually annotated as verbs ”1” or non-verbs ”0”. 1979 words are annotated as verbs while the remaining 2731 words are annotated as non-verbs. we call this dataset ”Stories”.

The second dataset is extracted from the Lisan corpus [16]. The Lisan dataset contains 1.5 million tokens from five Arabic dialects including 50K morphologically annotated LD tokens. Diacritics are used with tokens to differentiate between verbs, nouns and other parts of speech tokens. We extracted all LD tokens labeling verbs with ”1” and non-verbs with ”0”. The final dataset contains 9822 tokens annotated as verbs and 40740 tokens annotated as non-verbs. We call this dataset ”Lisan”. Both datasets are usually divided into 80% for training 20% for testing in all our experiments

### 3.2 Evaluation Measures

Most known metrics for evaluating machine learning algorithms in a classification task are Accuracy, Precision, Recall, and F1 Score. We use accuracy to show the model

overall correctness, precision to evaluate the quality of the model prediction, recall to assess the ability of the model to fetch positive instances, and F1 score to balance the precision and recall.

Given that TP representing the number of none-verbs identified correctly as verbs, FP is the number of words identified falsely as verbs, TN is the number of words identified correctly as none-verbs, and FN is the number of verbs identified falsely as none verbs; Accuracy, Precision, and Recall can be calculated as follows:

$$\text{Accuracy} = \frac{TP+TN}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$Precision = \frac{TP}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{TP}{\text{TP} + \text{FN}} \tag{3}$$

F1 Score is calculated using the precision and recall measures as follows:

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*\text{TP} + \text{FP} + \text{FN}} \tag{4}$$

There are few versions of F1 score, the best score was achieved by using the F1 micro score which is calculated by counting the total true positives, false negatives and false positives. We report our result using (A) to represent accuracy, (P) to represent precision, (R) to represent recall, and F1 to represent f1 scores.

### 3.3 Runs

2.3.1. Run 1: Using raw datasets.

We run these algorithms on the raw datasets without any text pre-processing as our dataset contains only words, although with diacritics. We did not remove duplicates. Table1 shows the results of running different algorithms on both datasets.

The results of the first experiment show that the Naive Bayes algorithm outperformed other algorithms in identifying verbs on the Lisan dataset scoring 0.832 on F1 measure, while the SVC algorithm was the best at identifying verbs using the Stories dataset at 0.633 on F1 measure. It is also evident that the Logistic Regression algorithm was best at recall scoring 0.658 and 0.621 on the Lisan and the Stories datasets respectively. The difference in the F1 score on both datasets is connected with several reasons, first, the Lisan dataset is bigger than the Stories dataset; second, the Stories dataset has no diacritics as it is collected from the social media websites, while the Lisan dataset contains diacritics as it has been prepared by professionals who differentiate between words according to their perspective position in the sentence using diacritics.

**Table 1.** Results obtained by running algorithms on both datasets without any pre-processing

| | LisanDataset | | | | Stories Dataset | | | |
|---|---|---|---|---|---|---|---|---|
| Algorithm | A | P | R | F1 | A | P | R | F1 |
| LR | 0.490 | 0.226 | **0.658** | 0.490 | 0.521 | 0.469 | **0.621** | 0.521 |
| DT | 0.626 | 0.273 | 0.548 | 0.626 | 0.512 | 0.435 | 0.345 | 0.512 |
| NB | **0.832** | **0.874** | 0.165 | **0.832** | 0.590 | 0.556 | 0.367 | 0.590 |
| NC | 0.645 | 0.277 | 0.507 | 0.645 | 0.521 | 0.447 | 0.345 | 0.521 |
| SVC | 0.785 | 0.438 | 0.351 | 0.785 | **0.633** | **0.607** | 0.484 | **0.633** |
| RF | 0.635 | 0.278 | 0.543 | 0.635 | 0.516 | 0.440 | 0.341 | 0.516 |

**Table 2.** Results obtained by running algorithms on Lisan dataset after removing duplicates.

| | Lisan Dataset | | | |
|---|---|---|---|---|
| Algorithm | A | P | R | F1 |
| LR | 0.523 | 0.369 | 0.650 | 0.523 |
| DT | 0.611 | 0.432 | 0.599 | 0.611 |
| NB | 0.661 | 0.481 | 0.455 | 0.661 |
| NC | 0.615 | 0.434 | 0.581 | 0.615 |
| SVC | 0.690 | 0.537 | 0.454 | 0.694 |
| RF | 0.615 | 0.435 | 0.593 | 0.615 |

Finally, the Lisan dataset has duplicates, while the Lisan has no duplicates. In the following runs, we will test the effects of removing diacritics, removing duplicates from the Lisan dataset, then we test the effects of removing verb suffixes on both datasets.

2.3.2. Run 2: Removing Duplicate Words.

In the second run, duplicate words in the Lisan dataset are removed before applying the different classification algorithms. This process left 18746 unique words with 6131 words annotated as verbs and 12615 word annotated as non-verbs. Table 2 shows the results obtained when algorithms are run on the dataset. The removal process decreased the effectiveness of all algorithms' classification, decreasing Naive Bayes F1 score from 0.832 to 0.661.

2.3.3. Run 3: Normalization and Removing Duplicates

Arabic text is usually written without diacritics except in children's or religious' books. As we target text on the social media, we removed diacritics. In this experiment, we applied normalization by removing any character that does not belong to the Arabic alphabet. We used the regular expression to remove any character that does not fall in range of [ي - ء]. After removing diacritics, we removed duplicates. This process left the

dataset with 10792 non-verbs and 5288 verbs. The remaining words are then split into 80% for training and 20% for testing. Table 3 shows the results of this experiment.

**Table 3.** Results obtained by running algorithms on the Lisan datasets after removing non-Arabic characters and removing duplicates.

| | Lisan Dataset | | | |
|---|---|---|---|---|
| Algorithm | A | P | R | F1 |
| LR | 0.514 | 0.362 | **0.667** | 0.514 |
| DT | 0.553 | 0.264 | 0.217 | 0.553 |
| NB | 0.649 | 0.455 | 0.438 | 0.649 |
| NC | 0.547 | 0.239 | 0.185 | 0.547 |
| SVC | **0.672** | **0.490** | 0.432 | **0.672** |
| RF | 0.559 | 0.264 | 0.205 | 0.559 |

Results show that the SVC algorithm performed the best scoring 0.672 on the F1 measure. However, the Logistic Regression algorithm scores the best Recall (0.667) among all algorithms. It is clear that this step has negatively affected the F1 score. This is expected as diacritics differentiate between verbs and non-verbs when having same letters. Removing diacritics would certainly negatively affect results.

2.3.4. Run 4: Normalization, Stemming, and Removing Duplicates

In this experiment, we stemmed certain suffixes from words in both datasets. As We are targeting verbs, we only removed pronoun suffixes that usually follow verbs. Particularly we removed "ت،اهو،مهمك،ان،ه،ك" if they exist at the end of any word. This step resulted in 9620 none-verbs and 4270 verbs in the Lisan dataset and 2408 none-verbs and 1706 verbs in the Stories dataset. The remaining text in the both datasets is then split into 80% for training and 20% for testing.

Table4 shows the results of running different algorithms. The SVC maintained its position as the best algorithm in identifying verbs when considering the F1 measure. However, the Logistic Regression is considered the best in terms of the Recall measure.

**Table 4.** Results obtained by running algorithms on both datasets after removing non-Arabic characters, stemming, and removing duplicates.

| | Lisan Dataset | | | | Stories Dataset | | | |
|---|---|---|---|---|---|---|---|---|
| Algorithm | A | P | R | F1 | A | P | R | F1 |
| LR | 0.523 | 0.351 | **0.700** | 0.523 | 0.470 | 0.394 | **0.605** | 0.470 |
| DT | 0.567 | 0.227 | 0.185 | 0.567 | 0.554 | 0.433 | 0.374 | 0.554 |
| NB | 0.694 | 0.489 | 0.472 | 0.694 | 0.581 | 0.470 | 0.377 | 0.581 |
| NC | 0.575 | 0.221 | 0.166 | 0.575 | 0.552 | 0.453 | 0.581 | 0.552 |
| SVC | **0.701** | **0.502** | 0.472 | **0.701** | **0.611** | **0.516** | 0.444 | **0.611** |
| RF | 0.573 | 0.230 | 0.181 | 0.573 | 0.552 | 0.428 | 0.359 | 0.552 |

## 4 Discussion

Results of different experiments show that verb identification in LD text is possible. Applying algorithms to the raw data gave the best result (0.832), we relate this to dataset size and word duplication. We suspect having duplicate words in the testset, causing the average macro F1 to be high. This was reflected when removing duplicates from the dataset. Having diacritic on words helps verb identification, this is true as Arabic language and the LD are full of words that are pronounced differently, although they contain the same letters and have different meanings. For example, the word ضَحِك /Dhahik/ (n.) meaning laugh, is similar to the word ضَحَكَ /Dhahaka/ (v.) meaning laughed, however diacritics make them different. Maintaining diacritics would improve algorithms' performance, however, LD text in reality is written without diacritics. Our experiments show that identifying verbs is possible using SVC at 70% on F1 measure. This percentage is high considering the nature of text and word ambiguity in the absence of diacritics. These experiments were run on a list of words without considering context or surrounding words in the sentence. We believe that having a special dataset where verbs are annotated within a text and using recent language models would improve verb identification.

## 5 Conclusions

In this paper, we tested six machine learning algorithms on identifying verbs in LD. Four experiments were run to check the effectiveness of these algorithms in recognizing verbs from other words in LD. Our results show that the Naive Bayes algorithm can be used when words contain diacritics, however, realistically SVC is the best algorithm to use to classify verbs from non-verbs in LD. Further research is required to identify verbs within a complete sentence.

## References

1. R. Alluhaibi, T. Alfraidi, M. A. R. Abdeen, and A. Yatimi, "A comparative study of arabic part of speech taggers using literary text samples from saudi novels," Information, vol. 12, no. 12, 2021.
2. N. Ramli, The Verb in Transitional Libyan Arabic: Morphomes, the Stem space and Principal parts. PhD thesis, University of Essex, October 2016.
3. R. Alharbi, W. Magdy, K. Darwish, A. AbdelAli, and H. Mubarak, "Part-of-speech tagging for Arabic Gulf dialect using Bi-LSTM," in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, eds.), (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.
4. O. Majewska and A. Korhonen, "Verb classification across languages," Annual Review of Linguistics, vol. 9, no. Volume 9, 2023, pp. 313–333, 2023.

5. S. Khoja, "Apt: Arabic part-of-speech tagger," in Proceedings of the Student Workshop at NAACL, pp. 20–25, 2001.
6. M. Diab, "Improved arabic base phrase chunking with a new enriched pos tag set," in Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, pp. 89–96, 2007.
7. R. Abumalloh, H. Muaidi, O. Ibrahim, and W. Abu-Ulbeh, "Arabic part-of-speech tagger, an approach based on neural network modelling,"International Journal of Engineering Technology, vol. 7, p. 742, 05 2018.
8. K. Darwish, "Building a shallow arabic morphological analyser in one day," in Proceedings of the ACL-02 workshop on Computational approaches to semitic languages, 2002.
9. M. Diab, "Towards an optimal pos tag set for arabic processing," in Proceedings of the International Conference on Recent Advances in Natural Language Processing, RANLP, pp. 157–161, 2007.
10. I. Zeroual, A. Lakhouaja, and R. Belahbib, "Towards a standard part of speech tagset for the arabic language," Journal of King Saud University - Computer and Information Sciences, vol. 29, no. 2, pp. 171–178, 2017. Arabic Natural Language Processing: Models, Systems and Applications.
11. K. Darwish, H. Mubarak, A. Abdelali, M. Eldesouki, Y. Samih, R. Alharbi, M. Attia, W. Magdy, and L. Kallmeyer, "Multi-dialect Arabic POS tagging: A CRF approach," in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, eds.), (Miyazaki, Japan), European Language Resources Association (ELRA), May 2018.
12. M. T. B. Othman, M. A. Al-Hagery, and Y. M. E. Hashemi, "Arabic text processing model: Verbs roots and conjugation automation," IEEE Access, vol. 8, pp. 103913–103923, 2020.
13. B. Azman, "Root identification tool for arabic verbs," IEEE Access, vol. 7, pp. 45866–45871, 01 2019.
14. H. A. Ahmed, Abdulmonem and A. R. Tosun, "Using roots and patterns to detect arabic verbs without affixes removal," IJCSNS International Journal of Computer Science and Network Security, vol. 23, no. 4, pp. 1–6, 2023.
15. K. Darwish, M. Attia, H. Mubarak, Y. Samih, A. Abdelali, L. M`arquez, M. Eldesouki, and L. Kallmeyer, "Effective multi dialectal arabic pos tagging," Natural Language Engineering (NLE), 2020.
16. M. Jarrar, F. A. Zaraket, T. Hammouda, D. M. Alavi, and M. Waahlisch, "Lisan: Yemeni, iraqi, libyan, and sudanese arabic dialect copora with morphological annotations," in The 20th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA), (Giza, Egypt), IEEE, December 2023.

# التعرف التلقائي على الافعال في اللهجة الليبية

عبدالسلام النويصري[1] ، نبيلة شنبر[2]

[1]كلية تقنية المعلومات جامعة طرابلس

[2]كلية العلوم والتقنية طرابلس

[1]a.nwesri@uot.edu.ly, [2]shinbir@tcst.edu.ly

**الملخص:** يعد التعرف التلقائي على الأفعال أمراً بالغ الأهمية للعديد من المهام التي تتعلق بمعالجة اللغة الطبيعية. تعرض الأفعال المعلومات العلائقية في الجملة بين الفعل واجزائها المختلفة وتعتبر المصدر الأساسي للمعلومات في فهم الجملة والأساس لأي مهمة تتعلق بمعالجة اللغة الطبيعية.

في هذه الورقة، قمنا بتجربة ستة خوارزميات للتعلم الآلي لتحديد الأفعال من الكلمات الأخرى في اللهجة الليبية. من بين الخوارزميات المستخدمة، كانت خوارزمية الـ (SVC) الأفضل في تحديد الأفعال بنسبة 70٪ لمقياس F1.

**الكلمات المفتاحية:** الافعال ، معالجة اللغات الطبيعية، تعلم الآلة، اللهجة الليبية.