Shami: A Corpus of Levantine Arabic Dialects

Kathrein Abu Kwaik[*], [†]**Motaz Saad,** [*]**Stergios Chatzikyriakidis,** [*]**Simon Dobnik**
[*]CLASP, Department of Philosophy, Linguistics and Theory of Science, Gothenburg University, Sweden
[†]The Islamic University of Gaza, Palestine
kathrein.abu.kwaik@gu.se, motaz.saad@gmail.com, stergios.chatzikyriakidis@gu.se, simon.dobnik@gu.se

Abstract

Modern Standard Arabic (MSA) is the official language used in education and media across the Arab world both in writing and formal speech. However, in daily communication several dialects depending on the country, region as well as other social factors, are used. With the emergence of social media, the dialectal amount of data on the Internet have increased and the NLP tools that support MSA are not well-suited to process this data due to the difference between the dialects and MSA. In this paper, we construct the Shami corpus, the first Levantine Dialect Corpus (SDC) covering data from the four dialects spoken in Palestine, Jordan, Lebanon and Syria. We also describe rules for pre-processing without affecting the meaning so that it is processable by NLP tools. We choose Dialect Identification as the task to evaluate SDC and compare it with two other corpora. In this respect, experiments are conducted using different parameters based on n-gram models and Naive Bayes classifiers. SDC is larger than the existing corpora in terms of size, words and vocabularies. In addition, we use the performance on the Language Identification task to exemplify the similarities and differences in the individual dialects.

Keywords: Dialectal Arabic, Levantine Dialect Corpus, Dialect Identification.

1. Introduction

Arabic is one of the five most spoken languages in the world; it is spoken by more than 422 million native speakers and used by more than 1.5 billion Muslims.[1] The situation in Arabic is a classic case of diglossia, in which the written formal language differs substantially from the spoken vernacular. Modern standard Arabic (MSA), which is heavily based on Classical Arabic, is the official written language used in government affairs, news, broadcast media, books and education. MSA is the lingua franca amongst Arabic native speakers. However, the spoken language (collectively referred to as Dialectal Arabic) varies widely across the Arab world.

The rapid proliferation of social media resulted in these dialects finding their way to written online social interactions. Dialects of Arabic differ widely among each other and depend on the geographic location and the socioeconomic conditions of the speakers. They are commonly categorized in five dominant groups: Maghreb (Libya, Tunisia, Algeria, Morocco and western Sahara), Egyptian (Egypt and some parts of Sudan), Levantine (Palestine, Syria, Lebanon and Jordan), Iraqi (Iraq) and Gulf (Qatar, Kuwait, Bahrain, United Arab Emirates, Saudi Arabia, Oman, Yemen) (Zbib et al., 2012). Further varieties can be identified, for example the Jordanian dialect can be split to Urban, Rural and Bedouin.

MSA and Dialectal Arabic share a considerable number of lexical, semantic, syntactic and morphological features, but there are several differences as well. For example, the word (ايش *āyš*) means "what" in Palestinian but in MSA (اي + شيء *āy + šyʾ*) means "what thing". The word (زرابي *zrāby*) in Moroccan means "carpets" and is a synonym of (سجاد *sǧād*) or (بساط *bsāṭ*) in MSA. On the other hand, the word (وقية *wqyh*) in Algerian means "3 kg" while it means "1/4 kg" in MSA and Levantine, i.e. the same word has different meanings in Algerian, Levantine, and MSA.

Most of the Natural language processing (NLP) resources for Arabic concern Modern Standard Arabic (MSA) (Diab, 2009; Maamouri and Cieri, 2002; Manning et al., 2014; Habash et al., 2009; Habash, 2010). However, using these resources for Dialectal Arabic (DA) is considered a very challenging task given the differences between them.

It should be noted that, even though the Levantine dialects look similar to each other when only written form rather than spoken form is taken into consideration, there are a number of differences. Some of these are summarized in Table 1. There are additional reasons why discriminating Levantine dialects in text is challenging. Some of these include:

- Lack of accent in writing. For example, a word like (كيفك *kyfk* / *how are you*) is used in all dialects, but its pronunciation varies by country.
- High similarity between the Palestinian and Jordanian dialects, **except in some key words.**
- The political situation. For example, we find many Palestinians leveling their dialect closer to Syrian and Lebanese, which makes Palestinian appear as an intermediate dialect.
- Prestige. For example, many Bedouins or Rural people adapt their dialect to Urban to clarify the conversation.
- Lack of linguistic description and resources. For example, Levantine dialects are frequently treated as one dialect despite their differences.

In this paper, we focus on Levantine dialects which are spoken in Palestine, Syria, Jordan and Lebanon and collect a corpus of their usage from social media which we label as *Shami*. This is the first joint corpus of these four dialects. The name *Shami* derives from شامي *šāmy*, which in MSA means *Levantine*. We evaluate the corpus through a Language Identification task.

[1] http://www.unesco.org/new/en/unesco/events/prizes-and-celebrations/celebrations/international-days/world-arabic-language-day-2013/

الآنَ أَفْضَلُ شَيءٍ أَنَ الشَبابُ كُلهمْ *ālʿāna ʾafḍalu šayʾ ʾana ālšabaābu kulahum* ظَهَرُوا عَلَى السَاْحَةِ *ẓaharuwā ʿalaā ālsaāḥati*	MSA
Now the best thing is that guys have appeared on the scene	English
هَلَأْ أَحْلَى شِي إِنْوْ الشَبابَ كِلْلون ضَهَرُوا عسّاحِه *halaʿa ʾaḥlaā šiy ʾinw ālšabaāba killuwn ḍaharuwā ʿsāḥih*	Lebanon
هسا أَحْلَى اِشِي إِنْوْ الشَّبَابْ هَلَّأْ *hsā ʾaḥlaā āišiy ʾinw ālšabaāb halaʿa* كُلُّهُمْ ظَهَرُوا عَسَّاحَة *kuluhum ẓaharuwā ʿasaāḥah*	Jordanian
هَلْجِيت أَحْلَى اِشِي إِنْوْ الشَّبَابْ *halǧiyt ʾaḥlaā āišiy ʾinw ālšabaāb* كُلُّهُمْ طِلْعُوا عَسَّاحَة *kulahum ṭilʿuwā ʿasaāḥah*	Palestinian
هَلَّأْ أَحْلَى شِي إِنْوْ الشَّبَابْ كِلْلُون بَيَّنُوا عسّاحِه *halaʿa ʾaḥlaā šiy ʾinw ālšabaāb killuwn bayanuwā ʿsāḥih*	Syrian

Table 1: little differences between the parallel sentence among Levantine Dialects

2. Related Works

There are two kinds of Arabic dialectal resources: (i) resources that group all Levantine dialects in the same group; (ii) resources where individual Levantine dialects are represented but none of them contains all of the dialects that we are interested in.

The Arabic Online Commentary (AOC) dataset presents a monolingual dataset rich with dialectal content from three dialects (Gulf, Egyptian, Levantine) (Zaidan and Callison-Burch, 2011). In the case of Levantine, data are extracted from Jordanian newspapers only. Zbib et al. (2012) build Parallel Levantine-English and Egyptian-English parallel corpora, consisting of 1.1M words and 380k words, respectively.

Meftouh et al. (2015) present PADIC (Parallel Arabic Dialect Corpus). It includes five dialects: two Algerian, one Tunisia and two Levantine (Palestinian and Syrian). Bouamor et al. (2014) present a multi-dialectal Arabic parallel corpus. The dataset consists of 2,000 MSA sentences in Egyptian, Syrian, Palestinian, Tunisian, Jordanian and English. Table 2 in section 3 illustrate the size of these corpora.

A preliminary work on a Palestinian dialect corpus is presented by Jarrar et al. (2014). It includes 5,836 Palestinian sentences with 43K words.

Diab et al. (2010) designed a set of search queries for each dialect (Egyptian, Iraqi, Maghrebi and Levantine) to harvest automatically dialectal content from large online resources like weblogs and forums. Levantine dialects (Palestinian, Syrian, Jordanian, Lebanese) were assumed to comprise a single Arabic dialect. The data cover three domains only: social issues, politics and religion.

A similar corpus to AOC is presented by Cotterell and Callison-Burch (2014). However, again, most of the Levantine data are from one Levantine dialect only, namely Jordanian (6k sentences).

Almeman and Lee (2013) build a multi-dialect text corpus by bootstrapping dialectal words. They then categorize the dialectal text into four main categories depending on geographical distribution (Gulf, Levantine, Egyptian and North Africa), giving 14.5M, 10.4M, 13M and 10.1M tokens respectively.

3. Shami Dialect Corpus (SDC)

In this section, we present our *Shami* Dialect Corpus (SDC) which contains only Levantine dialects.

Its most important characteristics are: a) it is the first Levantine dialect corpus that contains the largest volume of data separated as individual Levantine dialects compared to the previous corpora; b) it is not a crafted and also not a parallel corpus; it contains real conversations as written in social media and blogs; c) it is not confined to a specific domain; it includes several topics from regular conversations such as politics, education, society, health care, house keeping and others; d) unlike previous corpora, SDC has been created from scratch by collecting Levantine data through automatic and manual approaches.

SDC is organized in text files, where each file represents one dialect. We have another structure for the Language Identification task, where we have a sub folder for each dialect and each sentence is represented as a separate text file. The corpus and the associated code can be found here: `https://github.com/GU-CLASP/shami-corpus`.

3.1. Data Collection

3.1.1. Automatic Collection

We use the Twitter API streaming library (*Tweepy*)[2] to collect as many relevant tweets as possible. Firstly, we gather twitter IDs from public Levantine figures and hence their linguistic background is known. Secondly, we use *tweepy* to collect tweets and replies from these IDs. Each streaming was run until 9,999 tweets are reached each time. We further use *tweepy* to extract data according to geographical location.

3.1.2. Manual Collection

Given that various domains and topics are needed for our corpus, we also collected a part of SDC manually. We harvest the web and choose online dialectal blogs for public figures in Levantine countries. We also extract discussions and stories from forums. Overall, this gives us sentences of various lengths.

[2] http://www.tweepy.org/

3.2. Data Pre-processing

Special treatment is required in order to pre-process dialectal text in order to standardize it and to make it useful for NLP applications in order to avoid a large number of single instance tokens[3]. For this reason, We employ the following processing steps:

- Remove diacritics: Arabic text has several diacritics which mark the pronunciation of the words and sometimes the meaning. We remove these diacritics from the corpus, for example: ّ Tashdid, َ *a* Fatha, ً *an* Tanwin Fath.
- Automatic data collection extracts many words and letters that do not belong to Arabic dialects. For example, Lebanese texts contain a lot of French and Arabic text in Latin characters as well as special characters like (@, !!, ??), number and dates, emoticons, and symbols.
- Normalization: there is no standard orthography for Arabic dialects. We try to unify the writing style by normalizing the spelling. Unlike previous work that applies across the board normalization, which sometimes unintentionally changes the meaning of the words, we implement finer rules that work more reliably and preserve the semantic meaning of the text, for example:
  a) Aleph: we only convert Aleph with an accent أ *ʔa* to Aleph without an accent ا *ā* if it appears at the beginning of the word. This is because we want to mark the accent in other contexts in order to preserve the meaning of dialectal words. For example, this allows us to distinguish (’هلأ *hlʔa*’ / now) from (’هلا *hlā*’/ Hello), which otherwise would be indistinguishable in meaning.
  b) Alef Maqsora (ى *ā*) at the end of the word: in most processing steps the letter (ى *ā*) is converted to a (ي *y*), but we did not do so because a lot of words would change the meaning if we unified the characters. For example: (على *ʕlā* / on preposition) and (علي *ʕly* Ali / a personal name). If we change the letter (ى *ā*), this will affect the context of the sentence.
- Remove repeated characters: In colloquial written speech as well as in social networks, some letters are frequently repeated to indicate length (for example Waaaaaaw). In previous works, all duplicate letters are removed to one or two letters. Again, we have specified some finer criteria to specify the repetition based on its origin. Below is our steps to remove repeated letters:
  1. We extract all words containing repeated characters in MSA texts and keep them in a list.
  2. All words containing duplicate characters from the previous list are abbreviated to two characters.
  3. The rest of the characters are reduced to only one character, for example the repeating character و *w* in (مبروووووك *mbrwwwwwk* / congratulation) is converted to (مبروك *mbrwk*/ congratulation). These are all alternative spellings imitating spoken language.
  4. The conjunction letter (و *w* / and) is a special case in Arabic. Some people in colloquial dialects connect it with the next word without a space. We postulated that if the given word begins with more than one (و *w*), the first (و *w*) and the rest of the word are separated and the original word is processed according to the previous algorithm. Figure 1 describes the algorithm for reducing repeated characters.
- Finally, we homogenize individual parts of the corpus to make sure they are free from MSA sentences as well as other non-Levantine dialects. This is done manually by native speakers of individual dialects.

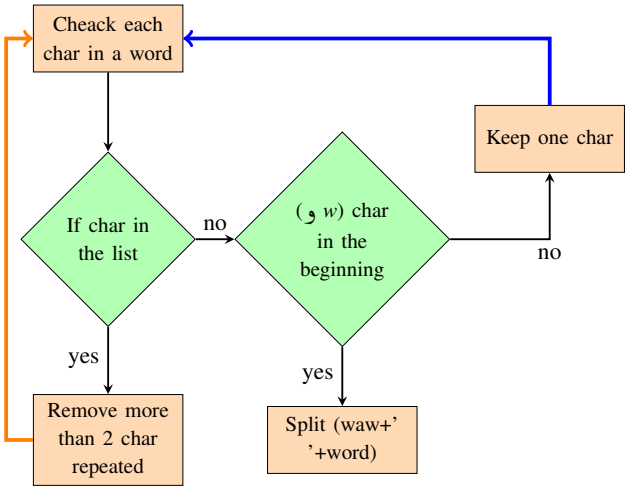

Figure 1: Algorithm for repeated characters in dialectal words

Compared with the PADIC (Meftouh et al., 2015) and the Multi-dialect Corpus (Bouamor et al., 2014), SDC is larger, more diverse, and more comprehensive. Table 2 reports the sentences, words, and vocabulary counts for SDC, PADIC and the Multi-dialect Corpus corpora.
In the next section, we use Arabic Dialect Identification as a test task for SDC in order to validate the corpus and test its usefulness. We then compare the results with PADIC and the Multi-dialect Corpus.

4. Arabic Dialect Identification

Arabic Dialect Identification can be performed at two levels: a) a coarse-grained level that builds a learner that given a specific Arabic sentence, measures the percentage of its dialectal content; b) a fine-grained level that classifies a sentence to the dialect in which it belongs.

[3]The original version of the corpus before pre-processing is also available if needed for other tasks

Shami Corpus			
	sentences	tokens	types
Jordanian	32078	3684369	85383
Palestinian	21264	2789103	69378
Syrian	48159	5268065	77918
Lebanese	16304	1409952	44418
Total	117805	13151489	227097
PADIC			
	sentences	tokens	types
Palestinian	6418	50827	22896
Syrian	6418	48701	27032
Total	12836	99528	49820
Multi-dialect Corpus			
	sentences	tokens	types
Jordanian	1000	9866	8905
Palestinian	1000	10315	8874
Syrian	1000	11586	9145
Total	3000	31767	26924

Table 2: Statistics for SDC, PADIC, and multi-dialects corpora

Zaidan and Callison-Burch (2014) train and evaluate an automatic classifier for Dialect Identification task on Maghrebi, Egyptian, Levantine, Iraqi and Gulf based on the corpus from Zaidan and Callison-Burch (2011) and they achieve an accuracy of 85.7% using a word-gram model. They conclude that using an n-gram word and character model is the most suitable method to distinguish among these dialects.

Elfardy and Diab (2013) propose sentence level identification and use words as tokens. They present a supervised classification method using Naive Bayes classifier to classify between MSA and Egyptian. They work on two parallel corpora. The first one is an Egyptian - Levantine - English corpus of 5M tokenized words of Egyptian (3.5M) and Levantine (1.5M). The second one is an MSA-English corpus with 57M tokenized words obtained from several LDC corpora. The system achieves different accuracy depending on the preprocessing steps and extracted features like percentage of dialect content, perplexity and metadata. The highest accuracy is 85% on the Arabic Online Commentary dataset AOC. In (Salloum et al., 2014), this work is extended to include the Iraqi, Levantine and Moroccan dialects.

Sadat et al. (2014) experiment with character n-grams using Markov Model and Naive-Bayes classifiers. The experiments are conducted on 6 main dialects defined by geographical area and 98% accuracy is achieved. However, the Levantine data constitutes the smallest dialect set and receives low overall accuracy.

Adouane et al. (2016) focus on Berber and various Arabic dialects (Algerian, Egyptian, Levantine, Gulf, Mesopotamian (Iraqi), Moroccan, Tunisian). They show that machine leaning (ML) models combined with lexicons are well suited for Dialect Identification as they achieve 93% accuracy when employed on 9 dialects whereby all Levantine dialects are grouped together.

A common characteristic of the previous Arabic Language Identification systems is that: (i) the data on which they are trained mostly comes from the same domain; (ii) most datasets and corpora are small in size; (iii) systems are trained and tested on different datasets with different parameters (size, domain, preprocessing).

Language Identification is a well-known task and given a sufficient amount of resources it can be considered a solved task. However, this does not hold for Arabic Dialect Identification. A lot of tools for Language Identification are available as open source, for example : *langid.py* (Lui and Baldwin, 2012) and *langdetect* (Shuyo, 2010). Given that the dialects that we are focusing on are very similar, our experiments may also give new insights with respect to Language Identification.

4.1. Langid.py for language identification

Lui and Baldwin (2012) present a tool for Language Identification called *langid.py*. In their tool they use Naive Bayes classifier with various n-gram character sequences for training purposes. The tool has been trained to identify 97 languages in the multi-domain Language Identification corpus of (Lui and Baldwin, 2011). The tool supports many modules so developers can easily train and build their own language models.

Comparing *Langid.py* to other Language Identification tools like *langdetect*, *TextCat* (Cavnar et al., 1994), and *CDL* (McCandless, 2011), Lui and Baldwin (2012) found that it is faster and gives better accuracy. This is why we also use *Langid.py* to conduct our corpus evaluation.

4.2. Scikit learn machine learning toolkit

Scikit learn (Pedregosa et al., 2011) is an open source python library that is a simple and efficient tool for data mining, data analysis and machine learning. It implements many machine learning methods like classification, regression and clustering and includes modules for preprocessing and feature selection. We use it to build Language Identification classifiers with word-gram models as *langid.py* does not support this.

5. Evaluation using Language Identification

Two techniques are commonly used in the literature for Language Identification. The first one is based on compiling lists of keywords for each language and scoring the text based on these lists (Richardson and Campbell, 2008). The second one uses Machine Learning such as Artificial Neural Networks (Al-Dubaee et al., 2010; Gonzalez-Dominguez et al., 2014), Support Vector Machines (Botha and others, 2008), Hidden Markov Models (Dunning, 1994) and N-gram models (Yang and Liang, 2010; Selamat, 2011) to identify languages.

We train two Dialect Identification systems a) a character based n-gram model with Naive Bayes classifier (*langid.py*); b) word based n-gram model with Naive Bayes classifier (*scikit learn*). We use n-gram based approaches because most of the variation between dialects can be identified by considering sequences of characters, e.g. affixation, and words, in addition to word features which can be provided by the lexicon. We conduct several experiments, which vary w.r.t the size of the data, the libraries

used (*langid.py* (Lui and Baldwin, 2012), *scikit learn* (Pedregosa et al., 2011)) and the classification techniques. For evaluation purposes, we measure the accuracy of the correctly identified test instances, while the F-measure gives us the balance between Precision and Recall.

5.1. Baseline system

To properly evaluate SDC's suitability as a corpus, we compare it to two Dialectal Corpora, i.e. PADIC (Meftouh et al., 2015) and Multi-dialect Corpus (Bouamor et al., 2014) using Language Identification. Firstly, we split the data and use 90% for training data and 10% for testing data. The first experiment was run with *Langid.py* with 4,5,6 and 7 n-character grams language model. Then, we evaluate these models using the test set. The second experiment was run with *scikit-learn* using uni-gram and bi-gram word models. The results for the two corpora are shown in Table 3.

PADIC			
	Language model	Accuracy	F-measure
langid.py	4-gram char	0.61	0.75
	5-gram char	0.64	0.78
	6-gram char	0.68	0.81
	7-gram char	0.68	0.81
Scikit-learn	uni-gram word	0.83	0.83
	bi-gram word	0.84	0.83
Multi-dialects Corpus			
	Language model	Accuracy	F-measure
langid.py	4-gram char	0.63	0.77
	5-gram char	0.68	0.81
	6-gram char	0.70	0.83
	7-gram char	0.69	0.82
Scikit-learn	uni-gram word	0.69	0.68
	bi-gram word	0.69	0.69

Table 3: Evaluation on PADIC and Multi-dialects Corpora

In general, the table shows that 6-gram models work best for Language Identification on the two corpora; it appears they are picking out particular phrases. In PADIC, the *scikit-learn* library with word-gram model outperforms *langid.py*. This is because PADIC is a parallel corpus of translated sentences where the differences are specifically emphasized when the corpus was built, and therefore more differences can be observed between Palestinian and Syrian. The Multi-dialect Corpus includes three Levantine dialects (Jordanian, Palestinian and Syrian), which makes the distinction between words harder, especially between Palestinian and Jordanian, which are very similar.

5.2. Dialect Identification with SDC

We carried out an experiment to determine the data size that will give us the highest performance. When we used all the data (Table 2), we get a low Accuracy ranging from 36% to 39% in all n-grams model as shown in Table 4.

As SDC is neither a parallel corpus nor a crafted corpus, many sentences are difficult to classify to a particular language. We commissioned four native speakers, one per each Levantine dialect, to create a subset of the data in order to reduce its heterogeneousness using the native knowledge of these dialect as shown in Table 5. Table 6 shows the results from training and testing on the filtered data from SDC. In comparison to raw data as shown in Table 4, the reduction of heterogeneousness improved the performance, thus, the classes of documents become more homogeneous in terms of the dialect.

	Language model	Accuracy	F-measure
langid.py	4-gram char	0.36	0.52
	5-gram char	0.38	0.53
	6-gram char	0.38	0.55
	7-gram char	0.39	0.55

Table 4: Evaluation on Sampling from SDC

Dialect	Train	Test	Total
Palestinian	9577	1065	10642
Syrian	33983	3776	37759
Jordanian	6316	702	7018
Lebanese	9747	1083	10830

Table 5: Train and test set for SDC after filtering

	Language model	Accuracy	F-measure
langid.py	4-gram char	0.54	0.70
	5-gram char	0.65	0.71
	6-gram char	0.55	0.71
	7-gram char	0.55	0.71
Scikit-learn	uni-gram word	0.70	0.71
	bi-gram word	0.70	0.70

Table 6: Evaluation on Sampling from SDC

However, the systems do not perform as well on SDC as on PADIC and Multi-dialect Corpus. This shows that Dialects Identification is more difficult in the case of SDC than the other two corpora. This is presumably because SDC is a more natural corpus than the other two which were translated with a focus on differences.

To confirm this, we have done a survey of several sentences without typical dialectal keywords and asked Levantine native speakers to classify each sentence to the dialect it belongs. For example, one of the sentences was الدرس اليوم كان كتير حلو وما حسينا بملل بالمرة. *āldrs ālywm kān ktyr ḥlw wmā ḥsynā bmll bālmrt.* ياريت كل يوم يكون هيك *yāryt kl ywm ykwn hyk* (The class today was very nice and interesting and we never felt bored. Hopefully every day will be like that). None of the participants could classify the Levantine dialect of this sentence with full certainty.

5.2.1. Comparing groups of dialects

Due to the great similarity between the Levantine dialects, we have conducted further experiments to compare subsets of dialects in the classification. We first used two-way classification between Palestinian and Syrian (as in the PADIC corpus) and between Jordanian and Lebanese (Table 7), and a three-way classification between Palestinian, Jordanian

and Syrian as in the Multi-dialect Corpus. Due to the similarity of Palestinian to the other dialects, we excluded it in the final classification, which included Jordanian, Syrian and Lebanese only (Table 8). Figures 2, 3 show the F-measure for the classification task for two and three dialects comparing with the PADIC and the Multi-dialect corpus.

Palestinian - Syrian Classification			
	Language model	Accuracy	F-measure
langid.py	4-gram char	0.73	0.83
	5-gram char	0.72	0.83
	6-gram char	0.72	0.84
	7-gram char	0.72	0.83
Scikit-learn	uni-gram word	0.87	0.85
	bi-gram word	0.80	0.74
Jordanian - Lebanese Classification			
	Language model	Accuracy	F-measure
langid.py	4-gram char	0.87	0.88
	5-gram char	0.89	0.89
	6-gram char	0.89	0.89
	7-gram char	0.89	0.89
Scikit-learn	uni-gram word	0.90	0.90
	bi-gram word	0.88	0.88

Table 7: Evaluation on two dialects classification

Palestinian - Jordanian - Syrian Classification			
	Language model	Accuracy	F-measure
langid.py	4-gram char	0.65	0.78
	5-gram char	0.65	0.79
	6-gram char	0.65	0.78
	7-gram char	0.64	0.78
Scikit-learn	uni-gram word	0.77	0.71
	bi-gram word	0.70	0.60
Jordanian - Syrian - Lebanese Classification			
	Language model	Accuracy	F-measure
langid.py	4-gram char	0.64	0.78
	5-gram char	0.81	0.82
	6-gram char	0.66	0.79
	7-gram char	0.65	0.79
Scikit-learn	uni-gram word	0.75	0.70
	bi-gram word	0.70	0.60

Table 8: Evaluation on three dialects classification

We get the highest performance when we classify Jordanian and Lebanese as they are to some extent different, and thus can be distinguished by text. Overall, this suggests that the Levantine dialects in SDC are very similar and therefore difficult to differentiate.

This is further emphasized by the fact that SDC is not a parallel corpus where textual differences between the sentences are specifically introduced as in the vase of PADIC and Multi-dialect Corpus. This way, the SDC highlights both similarities and differences between the four dialects. As it contains real data from social media and other sources as used by native speakers in everyday domains, it is a valuable resource for building NLP systems dealing with such data.

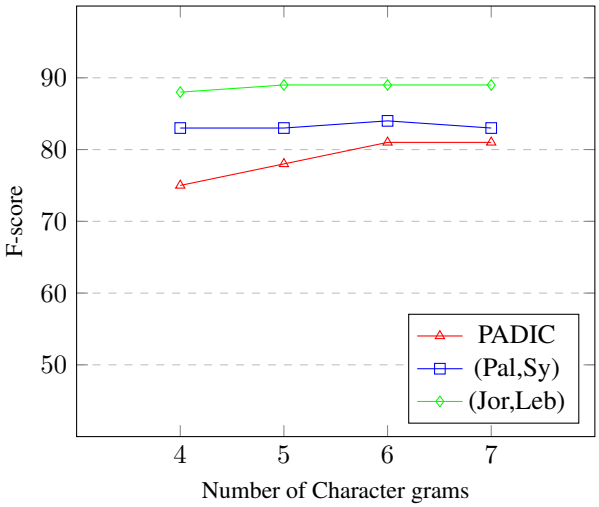

Figure 2: F-score on two dialect classification

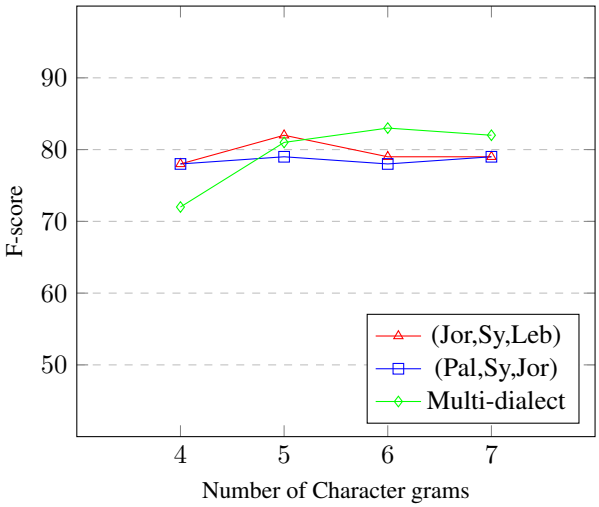

Figure 3: F-score on 3 dialect classification

6. Conclusions and Future Work

In this paper we present SDC, i.e. the first Levantine Dialect corpus which contains Palestinian, Jordanian, Syrian and Lebanese data. SDC contains natural data from new domains not available in the previous corpora. We have adopted a combination of manual and automatic methods to collect the documents. We then performed some preprocessing to standardize the spelling. Finally, we filtered the corpus to make it more homogeneous in terms of the dialects. We tested the usefulness of language models for Dialect Identification task by applying various n-gram models in two different classification approaches. At the same time, we compared the performance of Dialect Identification on SDC with that of PADIC and the Multi-Dialect Corpus corpora. The best results were achieved for classifying Jordanian and Lebanese only using a uni-gram word model (90% accuracy). This result is not surprising given a little similarity between the two dialects on the lexical level. The worst results were obtained when classifying for all four dialects on the whole SDC (52% accuracy). This is due to the great overlap between the dialects and dispersion of lexical items between categories.

We found that Language Identification on SDC outperforms other corpora when exactly the same dialects are considered. For example, Table 7 shows better performance than Table 3 for PADIC, and Table 8 shows better performance than Table 3 for Multi-Dialect corpus.

Our future works consist in extending the coverage of SDC and performing a corpus analysis of its linguistic features which will give us a more consistence picture of the differences between these dialects. The corpus will also be used to create lists of dialectal keywords. Finally, techniques other than Language Identification will also be tested on SDC.

7. References

Adouane, W., Semmar, N., Johansson, R., and Bobicev, V. (2016). Automatic Detection of Arabicized Berber and Arabic Varieties. *VarDial 3*, page 63.

Al-Dubaee, S. A., Ahmad, N., Martinovic, J., and Snasel, V. (2010). Language Identification Using Wavelet Transform and Artificial Neural Network. In *Computational Aspects of Social Networks (CASoN), 2010 International Conference on*, pages 515–520. IEEE.

Almeman, K. and Lee, M. (2013). Automatic Building of Arabic Multi Dialect Text Ccorpora by Bootstrapping Dialect Words. In *Communications, Signal processing, and their Applications (ICCSPA), 2013 1st international conference on*, pages 1–6. IEEE.

Botha, G. R. et al. (2008). *Text-based Language Identification for the South African Languages*. Ph.D. thesis, University of Pretoria.

Bouamor, H., Habash, N., and Oflazer, K. (2014). A Multidialectal Parallel Corpus of Arabic. In *LREC*, pages 1240–1245.

Cavnar, W. B., Trenkle, J. M., et al. (1994). N-gram-based Text Categorization. *Ann Arbor MI*, 48113(2):161–175.

Cotterell, R. and Callison-Burch, C. (2014). A Multi-dialect, Multi-Genre Corpus of Informal Written Arabic. In *LREC*, pages 241–245.

Diab, M., Habash, N., Rambow, O., Altantawy, M., and Benajiba, Y. (2010). Colaba: Arabic Dialect Annotation and Processing. In *Lrec workshop on semitic language processing*, pages 66–74.

Diab, M. (2009). Second Generation AMIRA Tools for Arabic Processing: Fast and Robust Tokenization, POS Tagging, and Base Phrase Chunking. In *2nd International Conference on Arabic Language Resources and Tools*, volume 110.

Dunning, T. (1994). *Statistical identification of Language*. Computing Research Laboratory, New Mexico State University.

Elfardy, H. and Diab, M. T. (2013). Sentence Level Dialect Identification in Arabic. In *ACL (2)*, pages 456–461.

Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., Gonzalez-Rodriguez, J., and Moreno, P. J. (2014). Automatic Language Identification Using Long Short-term Memory Recurrent Neural Networks. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Habash, N., Rambow, O., and Roth, R. (2009). MADA+ TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, volume 41, page 62.

Habash, N. Y. (2010). Introduction to Arabic Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Jarrar, M., Habash, N., Akra, D., and Zalmout, N. (2014). Building a Corpus for Palestinian Arabic: A Preliminary Study. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27.

Lui, M. and Baldwin, T. (2011). Cross-domain Feature Selection for Language Identification. In *In Proceedings of 5th International Joint Conference on Natural Language Processing*. Citeseer.

Lui, M. and Baldwin, T. (2012). Langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30. Association for Computational Linguistics.

Maamouri, M. and Cieri, C. (2002). Resources for Arabic Natural Language Processing. In *International Symposium on Processing Arabic*, volume 1.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford Corenlp Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

McCandless, M. (2011). Accuracy and Performance of Google's Compact Language Detector.

Meftouh, K., Harrat, S., Jamoussi, S., Abbas, M., and Smaili, K. (2015). Machine Translation Experiments on PADIC: A Parallel Arabic Dialect Corpus. In *The 29th Pacific Asia conference on language, information and computation*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Richardson, F. S. and Campbell, W. M. (2008). Language Recognition With Discriminative Keyword Selection. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 4145–4148. IEEE.

Sadat, F., Kazemi, F., and Farzindar, A. (2014). Automatic Identification of Arabic Dialects in Social Media. In *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*, pages 35–40. ACM.

Salloum, W., Elfardy, H., Alamir-Salloum, L., Habash, N., and Diab, M. (2014). Sentence Level Dialect Identification for Machine Translation System Selection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 772–778.

Selamat, A. (2011). Improved N-grams Approach for Web Page Language Identification. In *Transactions on*

Computational Collective Intelligence V, pages 1–26. Springer.

Shuyo, N. (2010). Language Detection Library for Java. *Retrieved Jul*, 7:2016.

Yang, X. and Liang, W. (2010). An N-gram-and-Wikipedia Joint Approach to Natural Language Identification. In *Universal Communication Symposium (IUCS), 2010 4th International*, pages 332–339. IEEE.

Zaidan, O. F. and Callison-Burch, C. (2011). The Arabic Online Commentary Dataset: An Annotated Dataset of Informal Arabic with High Dialectal Content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 37–41. Association for Computational Linguistics.

Zaidan, O. F. and Callison-Burch, C. (2014). Arabic Dialect Identification. *Computational Linguistics*, 40(1):171–202.

Zbib, R., Malchiodi, E., Devlin, J., Stallard, D., Matsoukas, S., Schwartz, R., Makhoul, J., Zaidan, O. F., and Callison-Burch, C. (2012). Machine Translation of Arabic Dialects. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 49–59. Association for Computational Linguistics.