LREC 2018 Workshop

OSACT 3:
The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools

PROCEEDINGS

Edited by

Hend Al-Khalifa, King Saud University, KSA
Walid Magdy, University of Edinburgh, UK
Kareem Darwish, Qatar Computing Research Institute, Qatar
Tamer Elsayed, Qatar University, Qatar

ISBN: 979-10-95546-25-2
EAN: 9791095546252

08 May 2018

Proceedings of the LREC 2018 Workshop
"The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools"

08 May 2018 – Miyazaki (Japan)

Edited by Hend Al-Khalifa, Walid Magdy, Kareem Darwish, Tamer Elsayed

http://edinburghnlp.inf.ed.ac.uk/workshops/OSACT3/

Organising Committee

- Hend Al-Khalifa, King Saud University, KSA
- Walid Magdy, University of Edinburgh, UK
- Kareem Darwish, Qatar Computing Research Institute, Qatar
- Tamer Elsayed, Qatar University, Qatar

Programme Committee

- AbdelRahim Elmadany, Jazan University, KSA
- Abeer Aldayel, King Saud University, KSA
- Ahmed Abdelali, Qatar Computing Research Institute, Qatar
- Ahmed Ali, Qatar Computing Research Institute, Qatar
- Ahmed Mourad, RMIT University, Australia
- Alberto Barrón-Cedeño, Qatar Computing Research Institute, Qatar
- Alexis Nasr, Université Aix Marseille, France
- Ali Jaoua, Qatar University, Qatar
- Almoataz B. Elsaid, Cairo University, Egypt
- Amal Alsaif, Al-Imam Muhammad ibn Saud Islamic University, KSA
- Ayah Zirikly, George Washington University, USA
- Azzeddine Mazroui, Université Mohammed Premier, Morocco
- Bassam Haddad, University of Petra, Jordan
- Eshrag Refaee, Jazan University, KSA
- Fahim Dalvi, Qatar Computing Research Institute, Qatar
- Fethi Bougares, Université du Maine, Avenue Laënnec, France
- Ghassan Mourad, Lebanese University, Lebanon
- Haithem Afli, Dublin City University, Ireland
- Hamdy Mubarak, Qatar Computing Research Institute, Qatar
- Hany Hassan, Microsoft, USA
- Hassan Sajjad, Qatar Computing Research Institute, Qatar
- Hassan Sawaf, Amazon, USA
- Hend Al-Khalifa, King Saud University, KSA
- Irina Temnikova, Qatar Computing Research Institute, Qatar

- Kareem Darwish, Qatar Computing Research Institute, Qatar
- Kemal Oflazer, CMU-Q, Qatar
- Khaled Bashir Shaban, Qatar University, Qatar
- Khaled Shaalan, The British University in Dubai, UAE
- Maha Alamri, Bangor University, UK
- Maha Althobaiti, Taif University, KSA
- Mahmoud El-Haj, Lancaster University, UK
- Mohammad Salameh, University of Alberta, Canada
- Mohsen Rashwan, Cairo University, Egypt
- Mucahid Kutlu, Qatar University, Qatar
- Muhammad Abdul-Mageed, The University of British Columbia, Canada
- Nada Ghneim, Higher Institute for Applied Science and Technology, Syria
- Nadi Tomeh, Université Paris 13, France
- Nasser Zalmout, New York University Abu Dhabi, UAE
- Nizar Habash, New York University Abu Dhabi, UAE
- Nora Al-Twairesh, King Saud University, KSA
- Preslav Nakov, Qatar Computing Research Institute, Qatar
- Salam Khalifa, NYU-AD, UAE
- Sarah Kohail, University of Hamburg, Germany
- Shady Elbassuoni, American University of Beirut, Lebanon
- Szymon Roziewski, Information Processing Institute, Warsaw, Poland
- Tamer Elsayed, Qatar University, Qatar
- Tim Buckwalter, University of Maryland, USA
- Violetta Cavalli-Sforza, Al Akhawayn University in Ifrane, Morocco
- Wajdi Zaghouani, Carnegie Mellon University, Qatar
- Waleed Ammar, Allen Institute for Artificial Intelligence, USA
- Wassim El-Hajj, American University of Beirut, Lebanon
- Younes Samih, Universität Düsseldorf, Germany

Preface

Given the success of the first and second workshops on Open-Source Arabic Corpora and Corpora Processing Tools (OSACT) in LREC 2014 and LREC 2016, where the presented papers received large number of citations, the third workshop (OSACT3) comes to encourage researchers and practitioners of Arabic language technologies, including computational linguistics (CL), natural language processing (NLP), and information retrieval (IR), to share and discuss their research efforts, corpora, and tools. In addition to the general topics of CL, NLP and IR, OSACT3 have given a special emphasis on a new Arabic Data challenge track.

OSACT3 had an acceptance rate of 65%, where we received 20 papers from which 13 papers were accepted. We believe that the accepted papers are high quality and present mixture of interesting topics. This year, we introduced ArabicWeb16, a new Web dataset that is suitable for many research projects. ArabicWeb16 is a public Web crawl of 150M Arabic Web pages, crawled over the month of January 2016, with high coverage of dialectal Arabic (about 21%) as well as Modern Standard Arabic (MSA). One goal of the workshop is to define shared challenges using this dataset. We encouraged submissions describing experiments for research tasks on the dataset. This includes (but not limited to) training word-embeddings, deduplication, cross-dialect search, question answering, dialect detection, knowledge-base population, entity search, blog search, text classification, and spam detection.

We would like to thank all people who in one way or another helped in making this workshop a success. Our special thanks go to Professor Mona Diab for accepting to give the workshop keynote presentation, to the members of the program committee who did an excellent job in reviewing the submitted papers, and to the LREC organizers. Last but not least we would like to thank our authors and the participants of the workshop.

H. Al-Khalifa, W. Magdy, K. Darwish, T. Elsayed — Miyazaki (Japan), 2018

Programme

Opening Session

09.00 – 09.10 Welcome and Introduction by Workshop Chairs

09.10 – 10.00 Mona Diab
Cross lingual modeling for low resource languages with a case application to Arabic Dialects (invited

10.00 – 10.30 Muhammad Abdul-Mageed
Learning Subjective Language: Feature Engineered vs. Deep Models

Session 1

11.00 – 11.20 Dima Taji, Jamila El Gizuli and Nizar Habash
An Arabic Dependency Treebank in the Travel Domain

11.20 – 11.40 Ramy Baly, Hazem Hajj, Wassim El-Hajj and Khaled Shaban
ArSentD-LEV: A Multi-Topic Corpus for Target-based Sentiment Analysis in Arabic Levantine Twee

11.40 – 12.00 Sawsan Alqahtani, Mona Diab and Wajdi Zaghouani
ARLEX: A Large Scale Comprehensive Lexical Inventory for Modern Standard Arabic

12.00 – 12.20 AbdelRahim Elmadany, Hamdy Mubarak and Walid Magdy
ArSAS: An Arabic Speech-Act and Sentiment Corpus of Tweets

12.20 – 12.40 Khaled Yasser, Reem Suwaileh, Abdelrahman Shouman, Yassmine Barkallah, Mucahid Kutlu and Tar
iArabicWeb16: Making a Large Web Collection More Accessible for Research

12.40 – 13.00 Jawad Sadek and Farid Meziane
Building a Causation Annotated Corpus: The Salford Arabic Causal Bank - Proclitics

Session 2

14.30 – 15.00 Areej Alshutayri and Eric Atwell
Creating an Arabic Dialect Text Corpus by Exploring Twitter, Facebook, and Online Newspapers

15.00 – 15.20 Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, Younes Samih and Mohammed Attia
Diacritization of Moroccan and Tunisian Arabic Dialects: A CRF Approach

15.20 – 15.40 Hamdy Mubarak
Dial2MSA: A Tweets Corpus for Converting Dialectal Arabic to Modern Standard Arabic

15.40 – 16.00 Abeer AL-Dayel, Hend Al-Khalifa, Sinaa Alaqeel, Norah Abanmy, Maha Al-Yahya and Mona Diab
ARC-WMI: Towards Building Arabic Readability Corpus for Written Medicine Information

Closing Session

16.30 – 17.00 Gilbert Badaro, Hussein Jundi, Hazem Hajj, Wassim El-Hajj and Nizar Habash
ArSEL: A Large Scale Arabic Sentiment and Emotion Lexicon

17.00 – 17.20 Wajdi Zaghouani and Anis Charfi
Guidelines and Annotation Framework for Arabic Author Profiling

17.20 – 17.30 Closing remarks

Table of Contents

ARLEX: A Large Scale Comprehensive Lexical Inventory for Modern Standard Arabic

Sawsan Alqahtani[1,3], Mona Diab[1], Wajdi Zaghouani[2]
[1]The George Washington University , [2]Hamad Bin Khalifa University, [3]Princess Nora Bint Abdulrahman University
[1]Washington DC, USA, [2]Doha, Qatar, [3]Riyadh, Saudi Arabia
{sawsanq, mtdiab}@gwu.edu, wzaghouani@hbku.edu.qa

Abstract

This paper introduces a lexical resource, ARLEX, for Modern Standard Arabic (MSA) that explicitly lists ambiguity at the lexical and syntactic levels for each token. Arabic orthography is known for being underspecified for short vowels and other markers such as letter doubling and glottal stops, known as diacritics. This leads to further ambiguity in orthography with real impact on natural language processing (NLP) applications, not to mention readability and human language processing. We specifically target listing alternative ambiguous forms of words within and across the same part of speech (POS), namely where tokens with no specified diacritics may have multiple possible diacritized alternatives. The entries in this dictionary are constrained to five POS tags: verbs, nouns, adjectives, adverbs, and prepositions. A morphological analyzer and disambiguator is leveraged to generate the desired linguistic properties. The resulting inventory, ARLEX, is a large scale comprehensive resource of words, recording their degree of ambiguity at various levels with example usages. ARLEX could be most useful for NLP applications, pedagogical applications, as well as socio- and psycho-linguistic studies.

Keywords: Lexicon, Diacritization, Arabic, Ambiguity

1. Introduction

Language ambiguity is an inherent characteristic of natural languages, which refers to the phenomenon where an instance can be interpreted in multiple ways. Ambiguity is at the core of problems faced by natural language processing (NLP) applications. Although humans have the ability to resolve such ambiguity based on their prior knowledge and context, there are instances (sentences, words, etc) that require multiple readings to resolve within context. The problem of natural language ambiguity is further exacerbated by conventional orthographic decisions where not all phonemes are explicitly represented.

Arabic standard orthography is one of these languages that is underspecified for such phonemes such as short vowels, gemination, etc, which are collectively represented as diacritic marks, aka diacritics. In other words, diacritics are crucial in denoting both pronunciations as well as meanings of such underspecified words. Most typical text in Arabic is rendered undiacritized, i.e. missing explicit diacritics, thereby compounding the linguistic ambiguity of text as observed, for instance, during the annotation of the various text types within the Qatar Arabic Language Bank project (Zaghouani et al., 2014; Zaghouani et al., 2015; Zaghouani et al., 2016b).

Orthographically fully specified Modern Standard Arabic (MSA) would consist of letters (consonants and long vowels) as well as diacritics. Diacritics can be divided into lexical, which specify the meanings of the words, and inflectional, which are added to provide syntactic roles of the words including syntactic case and mood endings as well as passivation. They comprise the short vowels (ـَ ـُ ـِ),[1] gemination marks (ـّ), nunation (ـً ـٌ ـٍ) denoting indefiniteness markers, and the absence of vowels (ـْ) typically used as a syllable delimiter as well as a mood marker. Because our concern is the meaning of the words, we only consider the internal diacritization (lexical) in this inventory and do not include syntactic case or mood diacritics nor general tanween (i.e. nunation) except where they are frozen, not syntactically motivated.[2]

A resource that lists words in their typical underspecified form and their corresponding possible meanings are useful for multiple purposes such as evaluating/building NLP tools, psycho-linguistic and socio-linguistic studies, as well as pedagogical applications.

In this paper, we present a monolingual large scale comprehensive lexical resource for MSA, ARLEX, which provides for each undiactrized word: various possible diacritized alternatives, together with other relevant information including: part of speech (POS), frequency of usage, genre usage, in addition to usage examples. It is a large scale automatically acquired inventory of words from multiple genres. The main objective of this inventory is to explicitly mark undiacritized forms of Arabic words when they are ambiguous. ARLEX represents different aspects of ambiguity at the word level: POS (syntactic level) and diacritized alternatives (lexical level). At the syntactic level, ambiguity indicates that the undiacritized word can have multiple possible POS tags. If an undiacritized word has a single POS then it is syntactically unambiguous. Within a given possible POS tag for an undiacritized form, a word may be lexically ambiguous as it may have multiple readings due to either multiple possible diacritizations or the same dia-

[1]We adopt the Buckwalter Transliteration (Buckwalter, 2002) system in the inventory.

[2]It is also worth noting that the diacritics may also include glottal stops, elongation, dots on letters, emphatic markers, or any additional normalization for the text such as replacing ا with آ, أ or إ where appropriate. However, we do not include them in this study.

critized form would have multiple meanings (similar to the bank 'financial institution' /bank 'river bank', in English). We account for all three ambiguity cases in ARLEX.

We use the morphological analyzer and disambiguator, MADAMIRA v1 (Pasha et al., 2014), to generate the desired features: POS, diacritized alternatives, and lemmas. It is important to note that ARLEX is not manually evaluated but rather uses human annotation in its development; crucially, it is tapping into the underlying morphological analyzer SAMA. Additionally, where available, we link entries in ARLEX with Tharwa (Diab et al., 2014). Tharwa enriches ARLEX diacritized lemmas with sense information as well as information such as meaning correspondents in dialects as well as English. Thus, ARLEX provides complementary information found in Tharwa and morphological analyzers such as SAMA.

2. Related Work

The phenomenon of language ambiguity has been investigated previously in several studies (Zaghouani et al., 2016; Versley, 2006). Zaghouani et al. (2016) provide linguistic analysis for possible ambiguity effects in MSA and show that automatic identification of ambiguous words helps reduce the annotation time. They ask annotators to tell whether they agree with the automatic ambiguity identification and then add missing diacritics to ambiguous words. Maamouri et al. (2012) created an educational tool and a corpus for Arabic reading enhancement by adding the diacritics to avoid the issue of word reading ambiguity.

In the the optimal diacritization scheme for Arabic orthographic representation (OptDiac) project (Bouamor et al., 2015; Zaghouani et al., 2016a), the focus was to create a large-scale annotated corpus with the diacritics for a variety of Arabic texts covering more than 10 genres to describe Arabic word pronunciation, and to create a valuable resource that can help address the issue of word reading ambiguity in the Arabic language.

Several lexical resources are available that help other research build and design their studies about languages (Zaghouani, 2014). This includes CELEX, Tharwa, AMPN, and SAMA. ARLEX is in line with such resources.

CELEX (Baayen et al., 1995) is a lexical resource that provides linguistic information for three languages: English (160,595 words), Dutch (381,292 words), and German (365,530). It compiles available manually annotated sources to provide detailed information about orthography, phonology, morphology, syntax, and frequencies at lemma and word levels. This resource is helpful for disambiguating the word forms since we may find multiple entries for the same word with slightly different information. ARLEX shares a subset of the objective presented in CELEX. CELEX, however, does not exist in Arabic.

Tharwa (Diab et al., 2014) is a multilingual lexicon that addresses the gap between different languages: English, MSA, and Arabic dialects with a current focus on Egyptian, Iraqi, Levantine. The publicly released Tharwa lexicon comprises 29,329 MSA, English, and Egyptian parallel instances. It is compiled to provide different linguistic information and help further studies in theoretical and computational linguistics. Although Tharwa provides a large repository of information about Arabic, it does not provide all possible alternatives for a given word as one of its objectives. The current proposed repository is an augmentation step to Tharwa where we link both resources using the index of MSA lemma and identify whether a diacritized lemma along with its POS has more than one sense.

AMNP (Hawwari et al., 2013; Zaghouani et al., 2016c) is a lexical semantic resource for Arabic morphological patterns. They built the morphological patterns' database using linguistic generalization of the semantic roles of the verbal predicates in the Arabic PropBank (Diab et al., 2008; Zaghouani et al., 2010; Zaghouani et al., 2012), which is a semantically annotated corpus of text from the Annahar Journal.

SAMA (Maamouri et al., 2010) is a morphological analyzer of MSA which provides all possible combinations of prefix, stem, and suffix for a given word. It also provides diacritization, clitic splitting information, and POS tags for each morpheme segment. SAMA maintains compatibility tables that show the appropriate combinations of prefix, stem, and suffix in MSA. This allows for the divination of all possible analyses for each given word. It includes 1,328 prefixes, 945 suffixes, and 79,318 stems. ARLEX is built on top of SAMA as MADAMIRA leverages it to provide all possible analyses and combinations as a first step in the disambiguation process. Our findings depend on SAMA output.

3. Dataset and Preprocessing

We use two datasets: the Arabic TreeBank (ATB) (Maamouri et al., 2008).[3] and the Contemporary Corpus of Arabic (CCA) (Al-Sulaiti and Atwell, 2006). ATB includes three genres: newswire (NW), broadcast news (BN), and web blogs (WB); CCA includes autobiography, children stories, economics, education, health medicine, interviews, politics, recipes, religion, science, short stories, sociology, spoken, and tourist travel. For preprocessing, we split all sentences in CCA at the punctation sentence periods.[4] Moreover, we leverage a dialectal identification tool, AIDA v2, to filter dialectal sentences (Al-Badrashiny et al., 2015), especially from the WB data in the ATB. Table 1 shows the number of sentences and words in the undiacritized forms for each genre, which also include numbers and punctuation. It is worth noting that ARLEX entries are in surface form as they occur in naturally occurring text with no preprocessing, which is different from SAMA and Tharwa where the entries are indexed by lemma form.

The ATB dataset provides human annotation for diacritics, POS tags, and lemmas for each undiaciritzed word. We use this information in our lexicon and complement them with automated information to construct a comprehensive lexicon as much as possible in terms of including all possible choices of alternative linguistic information. We apply the morphological analyzer and disambiguator, MADAMIRA (Pasha et al., 2014), to generate such alternatives for each

[3] Distributed by the Linguistic Data Consortium (LDC)

[4] CCA corpus includes long paragraphs; thus, we split the corpus by period which is the natural ending punctuation in most texts and fits our objective which is reducing the length of sentences.

Genre	# Sentences	# Words	Vocabulary Size
CCA	16,076	818,990	85,288
NW	23,488	630,634	65,404
BN	17,673	287,825	40,646
WB	3,818	58,468	18,222
TOTAL	61,055	1,795,917	142,381

Table 1: Corpus data statistics by genre indicating word types and word instances, where tokens are surface forms.

undiacritized word. MADAMIRA is trained on SAMA analyzer, discussed in Section 2., to retrieve all possible analyses for a given word and then uses a supervised classifier and a language model to rank the suggested choices. MADAMIRA do not provide analysis for words that are not recognized by its system; hence, we do not consider the automated analysis for such words. Table 2 shows some statistics of words with no provided analysis per genre.

Genre	# Types	# Words	% of No Analysis	% of Genre
CCA	7,311	41,334	60.17%	5.05%
NW	5,127	14,859	21.63%	2.36%
BN	1,702	10,356	15.07%	3.60%
WB	8,961	2,143	3.12%	3.67%
Overall	23,101	68,692	100%	3.82%

Table 2: This table shows the number of types (unique surface forms of words) and the number of words with no provided analysis in total and per genre. In addition, it shows the percentages of the words with no provided analysis compared to the total number of no-analysis words as well as the total number of words of the corresponding genre.

For CCA, we do not have human annotation for POS tags and lemmas, so we consider the top choices generated by MADAMIRA as the correct choice despite the possible errors (i.e. equivalent to the human annotation in ATB). CCA provides human annotated diacritization on the majority of the words which accounts for 93.64% of the data. However, where there is no human annotation for diacritization, we also use MADAMIRA's top choice.

For cleaning, we remove case and mood related diacritics from the diacritized version of the corpus since it does not contribute to the lexical meanings. We restrict our inventory to have a closed set of POS tags which are verbs, nouns, adverbs, adjectives, and prepositions.We do not accept any word normalization.[5]

MADAMIRA reports an accuracy of 95.9% for POS tagging and 86.3% in diacritization where both gold (i.e. humanly annotated) and automated words being compared have to be an exact match in tokenization, spelling, and full diacritization including syntactic case and mood markers. Alqahtani et al. (2016) evaluates MADAMIRA performance in diacritization in BN and WB genres, which are not used in MADAMIRA's training phase. They report 90.65% accuracy for full diacritization and 96.38% in full diacritization without case and mood diacritics which is the one we are using in the current work.

[5]MADAMIRA suggests alternative normalization variants for the following three groups (إ ء أ ا آ), (ى ي), (ة ه), as a result of anticipated spelling errors. For example, one of the suggestions for the word "أرجاء" [all around] in MADAMIRA is to convert it to "إرجاء" [postpone] which both have very different meanings.

4. Inventory

For each given undiacritized word in the corpus, we compile a list of relevant lexical information which are helpful in studies that concern lexical ambiguity in addition to potentially finding interesting relationships between ambiguity and other parameters. This lexicon is tab-serparated where each record contains the following information for each undiacritized word:

- `UNDIAC`: the surface word space-tokenized without any diacritic marks, i.e. undiacritized word (e.g. "الكتب" or "كتبه");
- `DIAC`: a possible diacritization for UNDIAC. (e.g. "الكُتُب" [the books] or "أَكْتُبه" [I am writing it]);
- `UNDIAC and DIAC LENGTH`: the number of characters in UNDIAC and DIAC forms;
- `UNDIAC TOKEN`: the core token/stem of the word without any prefixes and suffixes (e.g. the stem "كتب" for both "الكتب" and "أكتبه");
- `DIAC TOKEN`: the diacritized version of the UNDIAC TOKEN. This is useful to group words that have the same underlying meanings (e.g. the stem "كُتُب" for "الكُتُب" and "أَكْتُبه" for "أَكْتُب");
- `LEMMA`: the diacritized lemma of the word. This is also helpful to further specify the meaning of the word (e.g. the lemma "كِتاب" for "الكُتُب" and the lemma "كَتَب" for "أَكْتُبه");
- `POS`: the specific tag for DIAC as verb, noun, adjective, adverb, or preposition (e.g. "أَكْتُبه" is a verb and "الكُتُب" is a noun);
- `AMBIG POS`: For each UNDIAC TOKEN, 0 means that there is only one possible POS tag, and 1 means that there are multiple possible tags (e.g. 1 for "كتب");
- `AMBIG DIAC WITHIN POS`: For each UNDIAC TOKEN within an associated POS, 1 means that we have multiple diacritic alternatives, and 0 means that there is only one possible reading (e.g. 1 for "كتب" as a verb);
- `AMBIG DIAC`: For each UNDIAC TOKEN, 1 means that we have multiple diacritized alternatives within and across POS, and 0 means that there is only one possible reading (e.g. 1 for "كتب");
- `Tharwa Index`: the index of Tharwa lexicon that has the same lemma and POS as the current instance;
- `Tharwa Ambiguity Within POS And Diac`: Tharwa lexicon includes possible senses of the diacritized lemma along with its POS represented as possible English translations. For each LEMMA instead of TOKEN, 0 means that there is only one sense for the word within the same POS and DIAC according to Tharwa lexicon, and 1 means that there are multiple possible senses (e.g. 0 for "كَتَب" and "كِتاب");

- `UNDIAC_*, DIAC_*, DIAC_POS_*:` The symbol * refers to the a specific genre. These labels include information about the frequencies of UNDIAC, DIAC, and DIAC within the associated POS in each genre, respectively. In calculating such frequencies, we only consider the top choices within context for each word as it occurs in the corresponding gold ATB sentences and the top ranked POS tags and diacritics generated by MADAMIRA for CCA. We do not take into consideration the other possible alternatives provided by MADAMIRA;
- `TOTALs:` this set of values corresponds to the total frequencies of UNDIAC, DIAC, and DIAC within an associated POS in the whole corpus;
- `SENTENCES:` Representative example sentences from the corresponding corpora, which show the associated DIAC and POS (top analysis) in context. It is important to note that some records do not have associated examples because MADAMIRA provides all possible alternative choices which may not be present in the corpus as a top choice. For ATB, we use the gold diacritics and POS tags as the top choice. For CCA, we use the gold diacritics where available, for cases missing diacritics, and for all POS tags, we use the top choice generated by MADAMIRA.

  Each record is unique in terms of the diacritic variant, lemma, POS, and diacritized token such that deeper linguistic layers are available to use for researchers. Because we are combining gold and automated resources, we need to obtain the linguistic information which is not provided by the human annotation from the corresponding automated analysis. This includes the same diacritics, POS, and lemma in case of ATB records and the same diacritics in case of CCA records. Thus, we compare the gold information with its automated counterparts; if there is a match, we accept the remaining linguistic information in the automated analysis. If there is no match, we try to maximize the mapping by editing the diacritized words in gold and automated resources so they match each other.

  For this reason, words that starts with 'وَال*' are considered the same as the ones start with 'وال*' so we do not consider the presence of the diacritic on the first letter, which is not necessarily specified since it can be inferred from the determiner 'ال'. We also filter out analysis that are exactly the same across all linguistic information except the diacritic in the prefixes 'ب' or 'ل' where the diacritic 'إِ' is optionally added; for instance, the set of words ('بِتَجَنُّب', 'بتَجَنُّب') and ('لِسُقُوط' , 'لسُقُوط') are the same across all linguistic information except the diacritic in the prefix. Furthermore, we filter those analyses whose undiacritized forms of the words are different than the corresponding gold undiacritized to ensure that there is no normalization of the word of any kind.

  Additionally, we filter gold entries of the following lemmas: typo (e.g. لِاسعارالنفط which is missing a space to separate two valid words), dialect (e.g. بِأيد), transerr (e.g. الجهاز), foreign (e.g. لِلنت), and DEFAULT (e.g. the invalid words *وَظفارن or *لِفوش) to further ensure the validity of the words and its associated examples. We are aware that some words of such types are valid but given incorrect lemmas[6] because of their presence with incorrect surrounding context so that annotators provide them a generic lemma. However, we choose to remove them all to make the process systematic and maintain valid lemmas to make the entries of this lexicon grouped in meaningful way.

  To link between Tharwa and ARLEX, We follow similar approach to maximize their mapping in terms of lemmas and POS tags. The POS tag sets differ in those resources in addition to the possible disagreement in the choice of POS tag for some words such as noun and adjectives. Thus, for matching, if the POS tag in ARLEX is noun, we consider identifying entries in Tharwa in a specific order of matching; we retrieve the entry of the first encountered POS tag and neglect the remaining choices. In particular, we compare the noun tag to the following order of POS tags in Tharwa: noun, vbn (i.e. verb, past participle), adjective, noun+prop (i.e. proper nouns), pple_act, noun_num (i.e. cardinal number), and noun+quant (i.e. quantifiers). If the POS tag is verb, we consider verb and then modal. If the POS tag is adjective then we consider adjective, pple_act, pple_pass, noun, and then adj+elative (i.e. adjective comparative). Adverbs and prepositions are mapped with their counterparts only because we have not encountered disagreement.

  We first identify Tharwa records in terms of lemma and POS information. If available, we add the associated Tharwa index to ARLEX. If no record is found, we compare a lemma and its POS tag of ARLEX to the word rather than the lemma and its POS in Tharwa . We similarly add the associated Tharwa index if available; otherwise, we consider the Tharwa index as '-1' which means no match.

4.1. Inventory Statistics

The analyses in our resource are augmented using possible valid combinations in SAMA. The number of total records in ARLEX is 343,919 where each instance is unique in terms of dicritized word, diacritized token, diacritized lemma, Tharwa index,and POS. The number of entries correspond to gold information is 155,495; the remaining are generated automatically without considering context.

4.1.1. Surface Forms of Words

As mentioned, we only consider a closed set of POS tags. Table 3 shows the percentages of considered and discarded tokens with respect to the POS tag per genre. As can be observed, the set of POS tags we have chosen constitute a large portion of the dataset which means that the discarded amount is not significant especially that it includes non-verbal words (e.g. numbers and punctuations).

The longest undiacritized surface word is 17 characters whereas the average length is 5 characters. Standard deviation for the undiacritized length is around 1.73 which shows that most words are spread over the average. We have 23 characters for the longest diacritized word and 8 characters on average. Most of the diacritized words in the corpus are of length near the average since the standard deviation is approximately 2.44.

Table 4 shows statistics at the surface forms in each category and overall corpus. The number of unique undiacritized surface words along with its POS is 148,396 whereas the number of unique undiacritized surface words

[6]For instance, the word 'الصَبْر' which has the lemma 'DEFAULT' in the ATB corpus and means 'patience or endurance'.

Statistic	NW	BN	WB	CCA	All
# Considered tokens	74.18%	81.42%	73.42%	69.81%	73.32%
# Discarded tokens	25.81%	18.58%	26.58%	30.18%	26.67%

Table 3: This table shows per genre: 1. The percentage of the surface forms of the words with the considered POS tags. 2: The percentage of the discarded surface forms of the words which have the remaining POS tags. Both combined construct the full dataset.

Statistic	verbs	nouns	adjs	advs	preps	Total
UNDIAC	37,098	79,451	30,903	171	773	113,570
Percentage	32.67%	69.96%	27.21%	0.15%	0.68%	-
DIAC	97,156	114,912	38,668	226	850	229,529
Percentage	42.33%	50.06%	16.85%	0.1%	0.37%	-
DIAC Increase	61.82%	30.86%	20.08%	24.34%	9.06%	50.52%

Table 4: This table shows the number of unique surface forms of the word for each category and for the whole corpus.

regardless of the POS is 113,570. This accounts for approximately 24% overlap between the surface words of different POS categories. Noun is the dominant category which accounts for 69.96% of the unique undiacritized words. Verb and adjective are the following categories which account for almost half of the occurrences of noun in the undiacritized version. Adverb and preposition comprise significantly a much smaller portion of the whole lexicon.

These observations change when the surface words are rendered diacritized. The number of unique diacritized surface word along with their corresponding POS is 481,341 whereas the number of unique diacritized words is 229,529 which shows around 53% overlap between diacritized words across different POS categories. Nouns and verbs are the most frequent POS tags that occur in the lexicon; adjective follow them in rank with a considerable gap. Verbs are the most frequent POS category that have diacritic variations which accounts for 61% increase of the number of surface words. The remaining POS categories experience increase due to diacritic variations at 20% to 30% except the preposition which goes under 9% increase. Overall, We have around 50% increase due to diacritic variations in the whole corpus.

4.1.2. Tokens and Lemmas

Table 5 shows statistics regarding the token and lemma levels for each POS tag. We consider the main token of the word to reduce sparseness in the data and to further focus in the underlying meaning. The number of undiacritized tokens is reduced by 44,330 which is 39% reduction compared to the surface forms. Noun is still the dominant POS category in the undiacritized token followed by verb and adjectives. The diacritized version of token follow the same pattern as the diacritized surface words. Verb is the most affected category due to diacritics which undergoes double increase in size. Adjective is the following POS category which accounts for around 33% diacritic variations. Noun then follows them in rank whereas adverb and preposition do not increase at a considerable percentage.

The lemma of the word further reduces sparseness as it focuses in the main meaning. As we can see from Table 5, noun has the most variations of lemmas followed by verbs and adjectives with significant gap. The number of unique lemma along with its POS tag is 28,606. The number of

Statistic	verbs	nouns	adjs	advs	preps	Total
UNDIAC	26,769	42,670	18,740	126	503	69,240
Percentage	38.66%	61.63%	27.07%	0.18%	0.73%	-
DIAC	58,733	60,568	28,030	143	537	134,652
Percentage	43.62%	44.98%	20.82%	0.11%	0.40%	-
DIAC Increase	54.42%	29.55%	33.14%	11.89%	6.33%	48.59%
LEMMA	6,736	15,653	5,725	113	379	25,223
Percentage	26.70%	62.06%	22.70%	0.45%	1.50%	-

Table 5: This table shows statistics at the token and lemma levels.

lemmas that are not found in Tharwa is 14,082 which account for 49.23%, which is a considerably high percentage. We also target ambiguity at different levels: syntactic and lexical. For the syntactic level, we have 18,043 undiacritized stems that are ambiguous at the POS which accounts for around 26%. For the diacritic alternatives, we have 25,664 undiacritized stems that are ambiguous in terms of the diacritic variations. The number of undiacritized tokens that are ambiguous within the POS tags are 26,067. If we do not constraint the ambiguity within the POS tag, the number of ambiguous words increase which is 39,413 undiacritized tokens. The number of lemmas along with their POS tags that are ambiguous within the diacritics is 1,395 such that we include all lemmas in our lexicon even the ones that have no link to Tharwa. We do not have knowledge of the sense ambiguity within the diacritics of the remaining lemmas.

4.2. Discussion

The absence of diacritics adds an additional layer of ambiguity in MSA. Diacritics help specify the exact meanings or even reduce the number of possible senses for a given undiacritized word. Although this sounds appealing and has proven beneficial in some tasks (Vergyri and Kirchhoff, 2004; AlHanai and Glass, 2014), full diacritization might also have performance degradation in some NLP applications (Alqahtani et al., 2016; Diab et al., 2007) and human reading speed.

Maamouri et al. (2006) shows that there are three types of ambiguity caused by diacritics: ambiguity within POS tags, ambiguity for the same grapheme without considering POS tags, and ambiguity that is related to case and mood information. The former type concerns structural and grammatical level of ambiguity whereas the first two types are lexical which is our focus in this paper.

It has been claimed that frequency may play a significant role in disambiguation where words that frequently occur tend to be less ambiguous and that such frequency varies depending on the genre (Stokoe et al., 2003; Mihalcea et al., 2004; Lee and Myaeng, 2002). The current resource provides three types of frequencies: diacritized within a particular POS, undiacritized, diacritized words in addition to fine-grained frequencies for each genre so that researchers would be able to pick certain genres suitable for their studies. This lexical resource shows gaps in the frequency distributions among the alternative choices for each undiacritized word which may lead to having multiple choices for the same undiacritized word that have equal or close frequency approximation. This leads to an erroneous expectations which one must be careful about when having a limited-size data.For example, the word 'أصغر' can have

the following valid choices: 'أَصْغَر'[lesser/minimum/less] or 'أَصَغِّر'[I make something smaller] with frequencies 11 and 0, respectively.[7] This example clearly shows the significant difference in the frequency.

POS is an important factor that specifies the syntactic category of a word in a sentence (Ballesteros and Croft, 1998). It further helps refine the available diacritic alternatives for the undiacritized word and identify the specific meaning. For example, the word 'شطر' can be 'شَطْر'[bisector] as a noun and ''[sundered] or 'شَطَّرَ'[bisect] as verbs; there is no ambiguity within POS when the word is a noun because it can only take one form. On the other hand, the word is ambiguous in the case of verbs because it can take one of the two forms.

The main limitation of this resource is the automatic generation of linguistic information for each undiacritized word. In other words, we are relying on MADAMIRA for linguistic alternatives and have not evaluated this lexical resource through manual annotation. However, it is also costly and labor-intensive to create gold humanly-annotated lexical resource that provide all possible analysis and replace such a resource.

5. Conclusion

The main objective of this lexical resource is to help lexical-decision making based on explicitly marking within-POS ambiguity which means having multiple diacritic alternatives for the same undiacritized words within a particular POS. It also provides lexical information that is automatically generated including diacritic alternatives, POS, word length, frequencies (within and across varying corpora of different domains and genres) in addition to explicitly marking undiacritized words that have multiple possible POS, as well as providing usage examples. This resource will be used for readability experiments where we evaluate the impact of ambiguity and level of diacritization in human readings.

6. Bibliographical References

Al-Badrashiny, M., Elfardy, H., and Diab, M. (2015). Aida2: A hybrid approach for token and sentence level dialect identification in arabic. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 42–51.

Al-Sulaiti, L. and Atwell, E. S. (2006). The design of a corpus of contemporary arabic. *International Journal of Corpus Linguistics*, 11(2):135–171.

AlHanai, T. and Glass, J. (2014). Lexical modeling for arabic asr: A systematic approach. In *Proceedings of INTERSPEECH*.

Alqahtani, S., Ghoneim, M., and Diab, M. (2016). Investigating the impact of various partial diacritization schemes on arabic-english statistical machine translation. *AMTA 2016, Vol.*, page 191.

[7] In this example, we consider frequencies of the diacritized word constrained by a particular POS tag. 0 frequency means the surface form of the word never occurs in the corpus but it is a valid diacritic alternative for the word.

Ballesteros, L. and Croft, W. B. (1998). Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71. ACM.

Buckwalter, T. (2002). Arabic transliteration. *URL http://www.qamus.org/transliteration.htm*.

Diab, M., Ghoneim, M., and Habash, N. (2007). Arabic diacritization in the context of statistical machine translation. In *Proceedings of MT-Summit*.

Lee, Y.-B. and Myaeng, S. H. (2002). Text genre classification with genre-revealing and subject-revealing features. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 145–150. ACM.

Maamouri, M., Bies, A., and Kulick, S. (2006). Diacritization: A challenge to arabic treebank annotation and parsing.

Maamouri, M., Bies, A., and Kulick, S. (2008). Enhancing the arabic treebank: a collaborative effort toward new annotation guidelines. In *LREC*. Citeseer.

Maamouri, M., Zaghouani, W., Cavalli-Sforza, V., Graff, D., and Ciul, M. (2012). Developing aret: an nlp-based educational tool set for arabic reading enhancement. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 127–135. Association for Computational Linguistics.

Mihalcea, R., Tarau, P., and Figa, E. (2004). Pagerank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1126. Association for Computational Linguistics.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, pages 1094–1101.

Stokoe, C., Oakes, M. P., and Tait, J. (2003). Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 159–166. ACM.

Vergyri, D. and Kirchhoff, K. (2004). Automatic diacritization of arabic for acoustic modeling in speech recognition. In *Proceedings of the workshop on computational approaches to Arabic script-based languages*, pages 66–73. Association for Computational Linguistics.

Versley, Y. (2006). Disagreement dissected: Vagueness as a source of ambiguity in nominal (co-) reference. In *Ambiguity in Anaphora Workshop Proceedings*, pages 83–89.

Zaghouani, W., Hawwari, A., Alqahtani, S., Bouamor, H., Ghoneim, M., Diab, M., and Oflazer, K. (2016). Using ambiguity detection to streamline linguistic annotation. *CL4LC 2016*, page 127.

Zaghouani, W. (2014). Critical survey of the freely available arabic corpora. *International Conference on Language Resources and Evaluation, OSACT Workshop*.

7. Language Resource References

Baayen, R., Piepenbrock, R., and Gulikers, L. (1995). Celex2 ldc96l14. In *Web Download. Linguistic Data Consortium.*

Bouamor, H., Zaghouani, W., Diab, M., Obeid, O., Oflazer, K., Ghoneim, M., and Hawwari, A. (2015). A pilot study on arabic multi-genre corpus diacritization. In *Arabic Natural Language Processing Workshop, Association for Computational Linguistics Conference*.

Diab, M., Mansouri, A., Palmer, M., Babko-Malaya, O., Zaghouani, W., Bies, A., and Maamouri, M. (2008). A pilot arabic propbank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*.

Diab, M. T., Al-Badrashiny, M., Aminian, M., Attia, M., Elfardy, H., Habash, N., Hawwari, A., Salloum, W., Dasigi, P., and Eskander, R. (2014). Tharwa: A large scale dialectal arabic-standard arabic-english lexicon. In *LREC*, pages 3782–3789.

Hawwari, A., Zaghouani, W., O'Gorman, T., Badran, A., and Diab, M. (2013). Building a lexical semantic resource for arabic morphological patterns. In *Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on*, pages 1–6. IEEE.

Maamouri, M., Graff, D., Bouziri, B., Krouna, S., Bies, A., and Kulick, S. (2010). Standard arabic morphological analyzer (sama) version 3.1. *Linguistic Data Consortium, Catalog No.: LDC2010L01*.

Zaghouani, W., Diab, M., Mansouri, A., Pradhan, S., and Palmer, M. (2010). The revised arabic propbank. In *Proceedings of the Association for Computational Linguistics Fourth Linguistic Annotation Workshop*, pages 222–226. Association for Computational Linguistics.

Zaghouani, W., Hawwari, A., and Diab, M. (2012). A pilot propbank annotation for quranic arabic. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature co-located with the North American Association Computational Linguistics conference (NAACL-HLT 2012*, page 78.

Zaghouani, W., Mohit, B., Habash, N., Obeid, O., Tomeh, N., Rozovskaya, A., Farra, N., Alkuhlani, S., and Oflazer, K. (2014). Large scale Arabic error annotation: Guidelines and framework. In *International Conference on Language Resources and Evaluation (LREC 2014)*.

Zaghouani, W., Habash, N., Bouamor, H., Rozovskaya, A., Mohit, B., Heider, A., and Oflazer, K. (2015). Correction annotation for non-native Arabic texts: Guidelines and corpus. *Proceedings of The 9th Linguistic Annotation Workshop*, pages 129–139.

Zaghouani, W., Bouamor, H., Hawwari, A., Diab, M., Obeid, O., Ghoneim, M., Alqahtani, S., and Oflazer, K. (2016a). Guidelines and framework for a large scale arabic diacritized corpus. In *The Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3637–3643. European Language Resources Association (ELRA).

Zaghouani, W., Habash, N., Obeid, O., Mohit, B., and Oflazer, K. (2016b). Building an Arabic Machine Translation Post-Edited Corpus: Guidelines and Annotation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Portorož, Slovenia.

Zaghouani, W., Hawwari, A., Diab, M., OGorman, T., and Badran, A. (2016c). Ampn: a semantic resource for arabic morphological patterns. *International Journal of Speech Technology*, 19(2):281–288.

An Arabic Dependency Treebank in the Travel Domain

Dima Taji, Jamila El Gizuli, Nizar Habash
Computational Approaches to Modeling Language Lab
New York University Abu Dhabi, UAE
{dima.taji,nizar.habash}@nyu.edu

Abstract

In this paper we present a dependency treebank of travel domain sentences in Modern Standard Arabic. The text comes from a translation of the English equivalent sentences in the Basic Traveling Expressions Corpus. The treebank dependency representation is in the style of the Columbia Arabic Treebank. The paper motivates the effort and discusses the construction process and guidelines. We also present parsing results and discuss the effect of domain and genre difference on parsing.

Keywords: Arabic, Dependency, Treebank, Travel, Tourism

1. Introduction

Treebanks, or annotated corpora, are essential for Natural Language Process (NLP) tasks. Such tasks include building lexicons, inferencing grammars, and creating computational analyzers, which can all be improved by creating treebanks with different kinds of linguistic annotations (Abeillé, 2012). Treebanks with rich and good quality annotation are very expensive resources to create. They require a large number of man-hours to create and audit.

Treebanks can be in multiple genres, or genre-specific.[1] However, there is a tradeoff between the cost of the size, the diversity of a corpus, and having enough content in one genre or domain to be able to make generalizations. As a result, many treebanks tend to be predominantly of one specific genre, but may add some samples of other genres. For example, the Hindi/Urdu Treebank (Bhat et al., 2017) is predominantly in the news domain with 85.3% of its sentences coming from news articles, and only 14.7% from other domains (9.7% from conversations, and 5% from the travel domain). Webber (2009) shows that the Penn Treebank (Marcus et al., 1994) consists of 90.1% news articles, 4.9% essays, 2.6% summaries, and 2.4% letters, and it is still considered to be a news domain treebank. Similarly, Maamouri et al. (2010) demonstrate that the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) consists of 39.9% newswire text, 28.2% broadcast news, 18.6% broadcast conversation in both Standard and Dialectal Arabic, and 13.3% web texts.

In this paper we describe a small Modern Standard Arabic (MSA) treebank, created using a travel corpus. This treebank will be the seed of a larger multi-genre, and multi-dialect Arabic treebank. The corpus we are using is part of an MSA translation by Eck and Hori (2005) of the Basic Travel Expression Corpus (BTEC) (Takezawa et al., 2007), henceforth MSABTEC. As far as we know, there is no treebank based on this corpus.

[1]The terms *domain*, *genre*, *topic* and *style* have been discussed a lot in the field (Lee, 2002; Van der Wees et al., 2015; Ide and Pustejovsky, 2017), and many authors discussed their ambiguous and overlapping use. For the rest of this paper we use the term travel domain, following Takezawa et al. (2007) whose corpus was the basis for the translated corpus we treebank.

In Section 2, we discuss related work followed by a description of the corpus we annotate in Section 3. In Section 4, we discuss the annotation format; and in Section 5 the annotation process. Finally, we present some results on benchmarking parsing on our corpus and a comparison with a major news-domain Arabic treebank in Section 6.

2. Related Work

BTEC is a collection of conversational phrases that cover various situations in the travel domain in Japanese, and their translations into English and Chinese (Takezawa et al., 2007). The sentences in the corpus were collected from bilingual travel experts, and were based on their experience rather than being transcribed. The corpus was later translated into more languages including Arabic (Eck and Hori, 2005), where it was used for evaluating machine translation systems.

Another treebank that included phrases from the travel domain is the Hindi/Urdu treebank (Bhat et al., 2017). Even though the majority of the treebank comes from news sources, it contains 15K words, making up 1,058 sentences relating to heritage and tourism. This part of the data was specifically added to counteract the bias that could result from using data in one specific domain, news in this instance. The treebank contains dependency, phrase-structure, and PropBank-inspired (Kingsbury et al., 2002) annotations.

The Penn Treebank is a well known resource, that contains phrases mostly from the news domain. The treebank was annotated for genres as part of the Penn Discourse Treebank (Miltsakaki et al., 2004), and Webber (2009) shows that the different genres can have different characteristics.

The Penn Arabic Treebank (PATB) is the primary treebank for work on Arabic syntactic analysis. It uses a phrase-structure representation, but has been converted to other dependency formalisms (Habash and Roth, 2009; Taji et al., 2017). The PATB contains various parts that come from different domains and resources. PATB comes in 12 parts (Diab et al., 2013), that are mostly from news or web sources (Maamouri et al., 2010). Other related treebanks were developed by the Linguistic Data Consortium

(LDC) in various dialects such as Egyptian (Maamouri et al., 2012), and Levantine (Maamouri et al., 2006), where the data came from transcribing recorded conversations.

The first dependency Arabic treebank was the Prague Arabic Dependency Treebank (PADT) (Hajič et al., 2004). It employed a multi-level description scheme for functional morphology, analytical dependency syntax, and tectogrammatical representation of linguistic meaning.

Another large Arabic treebank is the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009). CATiB has around 250K words that were annotated directly in its dependency representation, in addition to a fully converted version of the PATB (PATB-CATiB). CATiB focuses on news domain text in Standard Arabic. Most recently, Taji et al. (2017) converted the PATB into the formalism of the Universal Dependency (UD) project (Nivre et al., 2016) via an intermediate step of mapping to CATiB dependencies.

The Quran Corpus is another important Arabic syntactic corpus of the very specific genre of holy scripture (Dukes and Buckwalter, 2010). It has its own representation scheme which is a hybrid dependency and constituency.

In this work, we annotate in the format of the CATiB treebank and compare to UD representations. And we present a comparison with the news domain as captured in the PATB.

3. Our Corpus

For our corpus, we selected the MSA translation of BTEC (Eck and Hori, 2005). Our selection contains 2,000 sentences making a total of 15,929 words (7.9 words/sentence). The sentences chosen are the same as those in CORPUS-25 from the Multi Arabic Dialect Applications and Resources (MADAR) project (Bouamor et al., 2018). The text of the corpus, coming from BTEC, is full of travel related expressions such as inquiring about the prices of hotel rooms, asking for directions, requesting help, ordering food, etc. Being conversational, it also has a high percentage of first and second person pronouns and conjugations. Below are examples of sentences from MSABTEC:

- أحتاج إلى طبيب. *ÂHtAj ĂlY Tbyb.*[2] 'I need a doctor.'
- كريمة وسكر؟ *krymħ wskr?* 'Cream and sugar?'
- أين أقرب محل جزارة؟ *Âyn Âqrb mHl jzArħ?*
  'Where is the nearest butcher?'

4. Annotation Format

To maximize compatibility with previous efforts, we followed the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009) annotation guidelines, and tokenization schemes used by previous Arabic treebanks. We chose this format because it uses traditional Arabic grammar as the inspiration for its relational labels and dependency structure (Habash and Roth, 2009), making it intuitive for Arabic speakers, and allowing for faster annotation. In addition, this format can be automatically enriched with more morphological features (Alkuhlani et al., 2013), and converted into other dependency formats such as the Universal Dependency format (Taji et al., 2017). Except for a number of minor specifications for some new syntactic constructions, there was no change to the guidelines for tokenization, part-of-speech (POS) tag set, and relations.

4.1. Tokenization

The tokenization followed in the treebank creation is the same tokenization scheme used in PATB. This scheme tokenizes all the clitics, except for the definite article +ال *Al+* 'the' (Habash, 2010). The 2,000 sentences in our corpus consist of 18,628 tokens (manually checked).

4.2. Annotation Scheme

For our treebank, we followed the CATiB dependency annotation scheme. This scheme is designed to be speedy for annotation, and intuitive for Arabic speakers. We also used the guidelines that were prepared for the CATiB annotation project (Habash et al., 2009).

4.2.1. POS Tags

The CATiB annotation scheme uses six POS tags which are **NOM** for all nominals excluding proper nouns, **PROP** for proper nouns, **VRB** for active-voice verbs, **VRB-PASS** for passive-voice verbs, **PRT** for particles, which include prepositions and conjunctions, and **PNX** for punctuation marks.

4.2.2. Relations

There are eight relations used in the CATiB scheme: **SBJ** for the subjects of verbs and the topics of simple nominal sentences; **OBJ** for the objects of verbs, prepositions, or deverbal nouns; **TPC** for the topics of complex nominal sentences which contain explicit pronominal referents; **PRD** for the complements of the extended copular constructions; **IDF** for marking the possessive nominal construction (*idafa*); **TMZ** for marking the *specification* nominal construction (*tamyiz*); **MOD** for general modification of verbs or nominals; and, finally, **—** for marking flat constructions such as first-last proper name sequences.

4.2.3. Syntactic Structures

Since the original CATiB treebank, as with the Penn Arabic treebank, was focused on the news genre, there were many syntactic constructions that MSABTEC introduced that needed special attention. In particular, there was an abundance of interrogatives, and first and second person statements in MSABTEC compared to CATiB. To address these constructions, additional guideline specifics and clarifications were added. All of these extensions followed naturally from the spirit of the original guidelines. For example, an interrogative pronoun such as من *man* 'who/whom' is often sentence-initial, but it can be the subject or the object of a verb: من سمع +ك؟ *man samiça +ka?* 'who heard you?' versus من سمعت؟ *man samiçta?* 'whom did you hear?'. Similarly, in Figure 1 (C), the interrogative adverb أين *Âyn* 'where' is treated as the predicate head of a copular sentence since that is the syntactic role of the answer to the question. For another common example in this genre, single word interjections such as آسف *Āsf* 'sorry'

[2]Arabic transliteration is presented in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007).

or شكراً *škrAã* 'thanks' are treated as independent sentence trees that attached directly to the main root of the sentence they appear in.

4.3. Interface

The annotation was done using the TrEd annotation interface (Pajas, 2008), which was also used by Habash and Roth (2009) for CATiB annotation.

Figure 1 illustrates the annotation scheme of three examples from the MSABTEC Treebank in the CATiB format in which they were annotated. We also provide, for comparison, the analysis in the increasingly popular Universal Dependency representation (Nivre et al., 2016; Taji et al., 2017).

5. Annotation Process

The annotation process we followed in the preparation of this treebank is the same process described by Habash and Roth (2009), which consisted of the following steps: *(a) Automatic Tokenization and POS Tagging*, *(b) Manual Tokenization Correction*, *(c) Automatic Parsing*, and *(d) Manual Annotation*. In this section, we discuss what we did for these steps as well as report on annotator(s), speed, and inter-annotator agreement.

5.1. Annotator(s)

Due to the relatively small size of our treebank, we had only one annotator working on the task. Our annotator is an educated native Arabic speaker, who was trained on the CATiB scheme and the use of TrEd as part of her work on the original CATiB project (Habash and Roth, 2009). To evaluate inter-annotator agreement, we worked with a second annotator who was asked to annotate a small part of the treebank (see below).

5.2. Automatic Tokenization and POS Tagging

We used MADAMIRA (Pasha et al., 2014) to tokenize and POS tag the input sentences. We used MADAMIRA's configuration for PATB tokenization and CATiB POS tags.

5.3. Manual Tokenization Correction

Our annotator then manually checked and fixed all of the tokenization errors. This also included the correction of typos and spelling changes resulting from wrong automatic analysis. Overall there were 2.8% tokenization errors, which is higher than MADAMIRA's reported tokenization error rate (around 1.1%). The increase is most likely due to the difference in genre between the data used to train MADAMIRA and our corpus.

5.4. Automatic Parsing

We ran the data with the fixed tokenization through the CamelParser (Shahrour et al., 2016), which is trained on the gold CATiB representation of the training data from the PATB parts 1, 2, and 3, according to the splits proposed by Diab et al. (2013). We present automatic parsing quality results in Section 6.2.

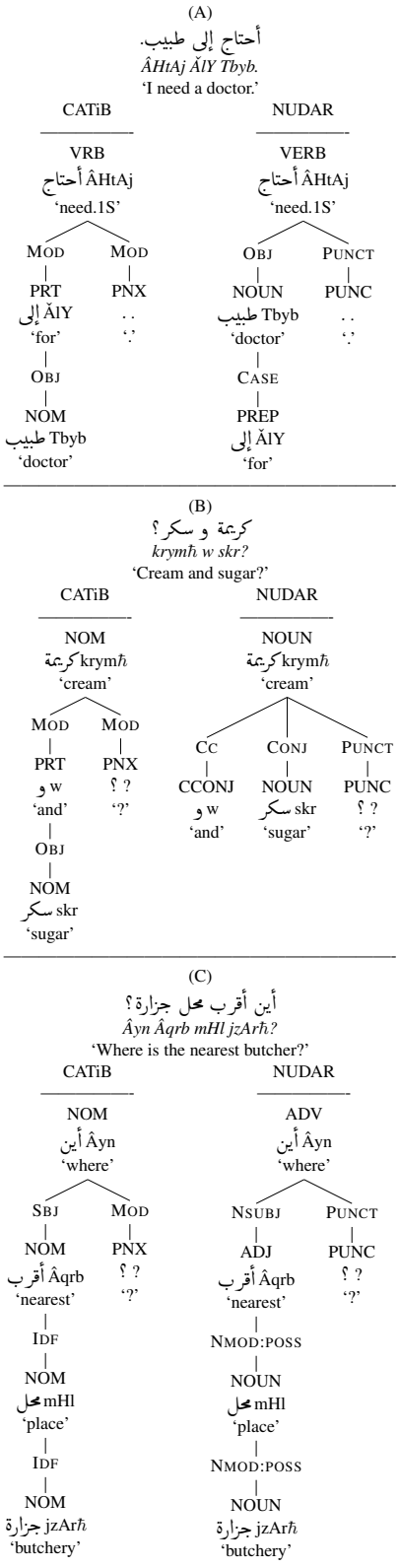

Figure 1: The structures for example trees from the MSABTEC Treebank in CATiB format, and their counterpart in the Arabic Universal Dependency (NUDAR) format (Taji et al., 2017).

5.5. Manual Annotations

The output of the automatic parsing was given in TrEd's *.fs* format to the annotator to manually fix the POS tags, the relation labels, and the syntactic structures of the trees.

5.6. Annotation Speed

The manual fixing of the tokenization took the annotator 10 hours of work at the speed of 1,593 words/hour. The manual correction of the parsed trees (POS, relations, and structure) took 40 hours of work at the speed of 466 tokens/hour (398 words/hour). This number is comparable to the speed reported by Habash and Roth (2009) (540 tokens/hour). The sentences in their treebank were of the same genre as the data used to train the automatic parsers unlike our case; furthermore, their sentences are much longer than ours (32.0 words/sentence compared to our 7.9 words/sentence). These two issues may explain part of the difference in speed. The end-to-end speed (from raw words to fully corrected trees) is 319 words/hour.

5.7. Inter-Annotator Agreement

To check the consistency of our annotations, we had another person with previous experience in dependency annotation annotate a subset of 100 sentences from this treebank. The second annotator started from the CamelParser output on the same corrected tokenization produced by the first treebank annotator. The inter-annotator agreement scores are 98.7% on POS agreement, 96.1% on label agreement, 90.6% on attachment agreement, and 89.7% on labeled attachment agreement. This is close to the highest average pairwise inter-annotator agreement number reported on the creation of the CATiB Treebank (Habash and Roth, 2009).

6. Results

We present next a comparison between our treebank and the Penn Arabic Treebank, followed by benchmark results of the performance of a state-of-the-art parser on our corpus. We use the abbreviation PATB to refer to facts about the content of the Penn Arabic Treebank, and PATB-CATiB to refer to the CATiB dependency representation of it.

6.1. Comparison with Penn Arabic Treebank

Our corpus is from the travel genre, which has some characteristics that are different from those of the news genre. For example, the average sentence length in MSABTEC is 7.9 words/sentence (9.3 tokens/sentence), as opposed to PATB's average of 32.0 words/sentence (37.6 tokens/sentence). Over 40% of MSABTEC sentences contained a question, while in PATB this percentage did not exceed 2.6%. This is expected as travel corpora are more likely to include questions and answers by travelers.

Moreover, the most frequent words in both corpora vary distinctly. MSABTEC's most frequent verb is يمكن *yumkin* 'can', which is often used when asking for help. In PATB, however, the most common verb is قال *qAl* 'said', which is commonly used for reporting news. In addition, question words such as كم *kam* 'how much', هل *hal* 'do/does', and اين *Âyn* 'where' appear in the set of the most frequent 50 words in MSABTEC, whereas no question words appear in the set respective to PATB. Frequent nouns in MSABTEC include فضل *faDl* 'favor/please', رقم *raqam* 'number', and غرفة *γurfaħ* 'room'. In PATB, the most frequent nouns include رئيس *raŷiys* 'president', لبنان *lubnAn* 'Lebanon', اليوم *Alyawm* 'today', and المتحدة *Almut∼aHidaħ* 'the united'.

Another phenomenon that differentiates MSABTEC and PATB is the pronoun frequencies. On the one hand, the most frequent pronouns appearing in MSABTEC are ك+ *+k*, which is the second person singular pronoun in accusative, and ي+ *+y* and ني+ *+ny*, which are the first person singular pronouns in genitive and accusative cases, respectively. On the other hand, the most frequent pronouns appearing in PATB are ه+ *+h* and ها+ *+hA*, which are the masculine and feminine third person singular pronouns, respectively. This leads to the obvious conclusion that MSABTEC mostly contains conversational text that refer to the speaker or the listener, whereas PATB's most dominant style is that of reporting in the third person, which is expected of a news genre corpus.

6.2. Automatic Parsing Quality

We compare the performance of a state-of-the-art parser on our MSABTEC corpus, against its performance on a standard test set from the same corpus it was trained on. The parser we are using is the CamelParser (Shahrour et al., 2016), which was trained and optimized on the PATB-CATiB corpus training set. The results are reported on the PATB-CATiB test set and the entire MSABTEC corpus. The evaluation metrics we are using are Labeled Attachment Score (LAS), Unlabeled Attachment Score (UAS), and Label selection, which measure the accuracy of the parser in predicting both the label and the parent, the parent only, and the label only, respectively.

	LAS	UAS	Label
PATB-CATiB test	83.8%	86.4%	93.2%
MSABTEC	73.5%	77.0%	90.5%

Table 1: The evaluation of the CamelParser prediction on data from PATB-CATiB test and MSABTEC

The error increase in the results of MSABTEC from the results on PATB-CATiB test for the LAS, UAS, and Label selection is 64%, 70% and 39%, respectively. This shows that the genre difference between the training data and the testing data significantly affects the performance of the parser. The previously described characteristics that differ between PATB and MSABTEC (sentence length, prevailing person, and different frequent words) can explain this decline in performance. The large performance drop highlights the need for creating treebanks in less-studied genres to support research on them.

7. Conclusion and Future Work

We presented a small dependency treebank of travel domain sentences in Modern Standard Arabic. The text comes from a translation of the English equivalent sentences in the Basic Traveling Expressions Corpus. The treebank dependency representation is in the style of the

Columbia Arabic Treebank. Our parsing evaluation of the constructed treebank confirms the need for more treebanks in different genres and domains to support research on multi-domain, multi-genre parsers.

In the future, we plan to expand our annotation efforts to other genres and domains as well as to other Arabic dialects. We are also very interested in using the created corpus in improving Arabic syntactic parsing. Since the data we created is small in size compared to the large dominant treebanks, we plan to pursue the genre and domain adaptation research direction. We also plan to make this resource publicly available to support research on Arabic syntactic parsing.

8. Bibliographical References

Abeillé, A. (2012). *Treebanks: Building and using parsed corpora*, volume 20. Springer Science & Business Media.

Alkuhlani, S., Habash, N., and Roth, R. (2013). Automatic morphological enrichment of a morphologically underspecified treebank. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 460–470, Atlanta, Georgia, June. Association for Computational Linguistics.

Bhat, R. A., Bhatt, R., Farudi, A., Klassen, P., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D. M., Vaidya, A., Vishnu, S. R., et al. (2017). The Hindi/Urdu treebank project. In *Handbook of Linguistic Annotation*, pages 659–697. Springer.

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., Obeid, O., Khalifa, S., Eryani, F., Erdmann, A., and Oflazer, K. (2018). The MADAR Arabic Dialect Corpus and Lexicon. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, May.

Diab, M., Habash, N., Rambow, O., and Roth, R. (2013). LDC Arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

Dukes, K. and Buckwalter, T. (2010). A Dependency Treebank of the Quran using Traditional Arabic Grammar. In *Proceedings of the 7th international conference on Informatics and Systems (INFOS 2010)*, Cairo, Egypt.

Eck, M. and Hori, C. (2005). Overview of the IWSLT 2005 evaluation campaign. In *International Workshop on Spoken Language Translation (IWSLT) 2005*.

Habash, N. and Roth, R. M. (2009). CATiB: The Columbia Arabic Treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224. Association for Computational Linguistics.

Habash, N., Soudi, A., and Buckwalter, T. (2007). On Arabic Transliteration. In A. van den Bosch et al., editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Habash, N., Faraj, R., and Roth, R. (2009). Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Habash, N. Y. (2010). *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Hajič, J., Smrž, O., Zemánek, P., Šnaidauf, J., and Beška, E. (2004). Prague Arabic Dependency Treebank: Development in Data and Tools. In *NEMLAR International Conference on Arabic Language Resources and Tools*, pages 110–117. ELDA.

Ide, N. and Pustejovsky, J. (2017). *Handbook of Linguistic Annotation*. Springer.

Kingsbury, P., Palmer, M., and Marcus, M. (2002). Adding semantic annotation to the Penn treebank. In *Proceedings of the human language technology conference*, pages 252–256. San Diego, California.

Lee, D. (2002). Genres, registers, text types, domains and styles: clarifying the concepts and navigating a path through the BNC jungle. *Language and Computers*, 42(1):247–292.

Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Maamouri, M., Bies, A., Buckwalter, T., Diab, M., Habash, N., Rambow, O., and Tabessi, D. (2006). Developing and using a pilot dialectal Arabic treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC Vol. 6*.

Maamouri, M., Bies, A., Jin, H., and Buckwalter, T. (2010). The Penn Arabic tree bank. *Computational Approaches to Arabic Script-Based Languages: Current Implementations in Arabic NLP. CSLI NLP Series*.

Maamouri, M., Bies, A., Kulick, S., Tabessi, D., and Krouna, S. (2012). Egyptian Arabic treebank pilot.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1994). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Miltsakaki, E., Prasad, R., Joshi, A., and Webber, B. (2004). The Penn discourse treebank. In *Proceedings of the Language Resources and Evaluation Conference*, Lisbon, Portugal.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, May. European Language Resources Association (ELRA).

Pajas, P. (2008). TrEd: Tree editor. http://ufal.mff.cuni.cz/ pajas/tred.

Pasha, A., Al-Badrashiny, M., Diab, M., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. M. (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *Proceedings of the Language Re-*

sources and Evaluation Conference (LREC), Reykjavik, Iceland.

Shahrour, A., Khalifa, S., Taji, D., and Habash, N. (2016). CamelParser: A system for Arabic syntactic analysis and morphological disambiguation.

Taji, D., Habash, N., and Zeman, D. (2017). Universal Dependencies for Arabic. *WANLP 2017 (co-located with EACL 2017)*, page 166.

Takezawa, T., Kikui, G., Mizushima, M., and Sumita, E. (2007). Multilingual spoken language corpus development for communication research. *Computational Linguistics and Chinese Language Processing*, 12(3):303–324.

Van der Wees, M., Bisazza, A., Weerkamp, W., and Monz, C. (2015). What's in a domain? Analyzing genre and topic differences in statistical machine translation. In *ACL (2)*, pages 560–566.

Webber, B. (2009). Genre distinctions for discourse in the Penn treebank. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 674–682. Association for Computational Linguistics.

ARC-WMI: Towards Building Arabic Readability Corpus for Written Medicine Information

Abeer Aldayel[1], Hend Al-Khalifa[2], Sinaa Alaqeel[3], Norah Abanmy[4], Maha Al-Yahya[5], Mona Diab[6]

[1,2,5]College of Computer and Information Science and [3,4]College of Pharmacy
King Saud University Riyadh, Saudi Arabia
{ [1]aabeer|[2]hendk|[3]salageel|[4]nabanmy|[5]malyahya }@ksu.edu.sa

[6]Department of Computer Science, the George Washington University
[6]mtdiab@gwu.edu

Abstract

Developing easy-to-read written medicine information continues to be a challenge in health communication. Readability aims to gauge the difficulty level of a text. Various formulas and machine learning algorithms have proposed to judge the readability of health materials and assist writers in identifying possible problems related to text difficulty. For this reason, having corpus annotated with readability levels is fundamental to evaluating the readability formulas and training machine learning algorithms. Arabic suffers from a lack of annotated corpora to evaluate text readability, especially for health materials. To address this shortage, we describe a baseline results towards constructing readability corpus ARC-WMI, a new Arabic collection of written medicine information annotated with readability levels. We compiled a corpus of 4476 sentences with over 61k words, extracted from 94 sources of Arabic written medicine information. These sentences were manually annotated and assigned a readability level ("Easy," "Intermediate," or "Difficult") by a panel of five health-care professionals.

Keywords: Corpus annotation, Readability corpus, Written medicine information

1. Introduction

Corpus annotation is the practice of adding metadata to a collection of text (Baker, 1997). These metadata relate to specific parts of the text (i.e., a word or a sentence) and are used to add both linguistic and descriptive information to it. Annotated corpora emerged to model various language phenomena and to train algorithms (Pustejovsky and Stubbs, 2012). There are different types of annotation tasks: one relates to linguistic models, such as a semantic annotated corpus (Basile et al., 2012) and a syntax annotated corpus (Brants et al., 2002), and the other relates to natural language processing (NLP) tasks, such as an inference corpus (Bowman et al., 2015).

Many recent studies have emerged to address the need for large health materials corpora with linguistic or NLP related metadata added to the text. There are health materials annotated corpora with sentiment-related information, such as clinical sentiment corpus (Deng et al., 2016), and other health related materials annotated with linguistics metadata, such as clinical part-of-speech tagging corpus (Pakhomov et al., 2006). Adding these metadata to health materials provides better insight into the data and facilitates obtaining robust results from the analyses.

Written medicine information (WMI) refers to the written information leaflet that accompanies medications (Koo et al., 2006). WMIs play an important role in educating consumers about their medicines. To contribute effectively to healthcare decision-making, these resources should be written at a level readable by any patient. Known as health literacy, measuring the readability for health related text is a long-established problem. Health literacy is defined as the degree to which individuals have the ability to understand basic health to make appropriate health decisions (Hewitt, 2012). Different methods have been used, such as traditional formulas and machine learning algorithms, to predict the text difficulty level and automatically predict the level of text readability. These methods need to be evaluated using a corpus annotated with readability levels (Koo et al., 2006).

In this paper, we introduce the ARC-WMI Arabic Readability Corpus. Comprised of more than 4000 sentences, it contains WMIs annotated with readability levels and collected from two sources: the Saudi Food and Drug Authority (SFDA)[1] and the King Abdullah Bin Abdulaziz Arabic Health Encyclopedia (KAAHE).[2] The ARC-WMI will address the need for a readability corpus to evaluate the readability algorithm and the formulas in the Arabic health domain.
This paper is organized as follows: Section 2 reviews the related work on the readability corpus field. Section 3 outlines the constructed corpus. Section 4 presents the methodology (in detail) that we used for the annotation process. The conclusion and future directions follow in Section 5.

[1] http://www.sfda.gov.sa/En/Pages/default.aspx

[2] https://www.kaahe.org/en/

2. Related works

Numerous studies have emerged to address the need for a gold standard corpus for readability assessments in the health field. There are two common methods for evaluating text readability: (1) direct evaluation and (2) pair-wise comparison. In the direct evaluation method, the annotator assigns absolute scores or labels that reflect the text difficulty and uses the resulting mean readability score as the overall text difficulty score. Many studies, such as the one by (Kandula and Zeng-Treitler, 2008), where they annotated 324 health documents with the readability level based on a 1–7 Likert scale, follow this method to annotate text for readability. Another study (Rosemblat et al., 2006), used the same method to evaluate the readability of 22 consumer health texts based on linguistic and stylistic features. In the pairwise comparison method, the annotator will compare between two texts and judge the relative readability score between them. However, in this study (Van Oosten and Hoste, 2011), they used a pair-wise comparison to evaluate the readability of a large corpus that contained domain-specific documents, manuals, and patient information leaflets.

There has been a significant amount of work on linguistic related corpora for Arabic text, including morphological segmentation (Dukes and Habash, 2010), punctuated corpora (Zaghouani and Awad, 2016), and in-depth work on sentiment corpora (Abdul-Mageed and Diab, 2012). In contrast, Arabic suffers from a shortage of well-formed readability corpus, especially for health related materials. In this paper, we construct a collection of readability annotated WMIs texts to describe an ongoing effort to fill this gap.

3. Corpus description

In the Arabic Readability Corpus for Written Medicine Information (ARC-WMI), the readability annotation was conducted at the sentence level in which selected sentences from each piece of WMI was evaluated based on three readability levels (Easy, Intermediate, and Difficult). A total of 4476 sentences and approximately 61k words were collected from 94 WMIs. The WMIs were collected from two sources: 47 WMI from the Saudi Food and Drug Authority (SFDA) and 47 WMI from the King Abdullah Bin Abdulaziz Arabic Health Encyclopedia (KAAHE) . Table 2 illustrates the word and sentence distributions in each source. These two sources have different text structures and use different subheadings and sections, as shown in Figure 2, which forced us to define the distribution of the sentences for each WMI. In our corpus, we designed a coding scheme to enable the unique and descriptive tag identification for the sentences of any of the WMIs**.**

The ID tag naming follows this pattern: "$Source(file_n)\_S_z\_(Sentence_t)$", where S_z indicates the section number for each destination source (KAAHE or SFDA), as shown in Table 1. We defined the annotation values for the tags as integer numbers related to the text difficulty level (1. Easy; 2. Intermediate; 3. Difficult).

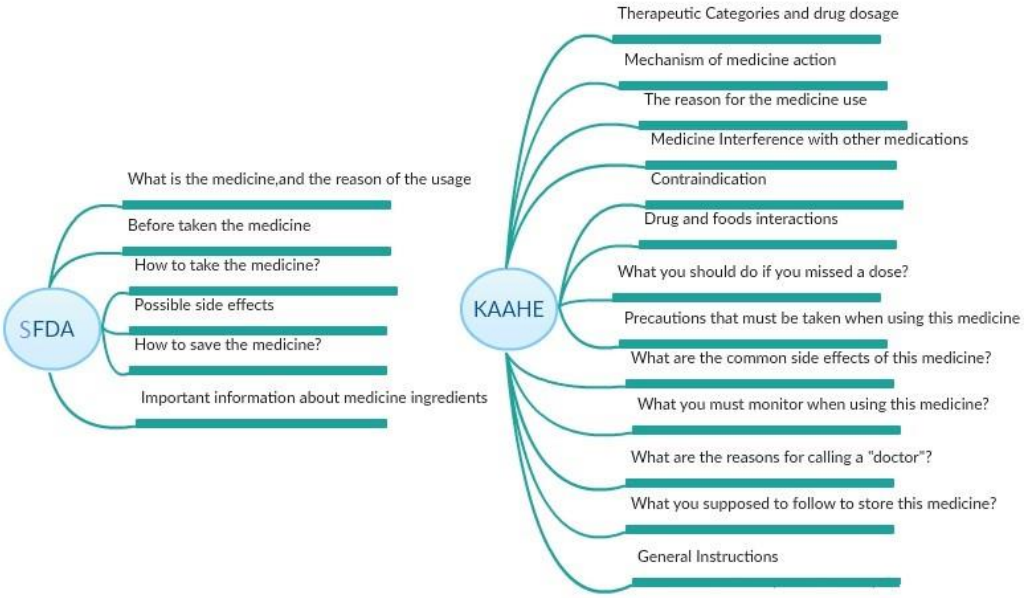

Figure 1 SFDA and KAAHE Structure

Table 1 Sentence coding

SFDA section (English)	SFDA section (Arabic)	Sentence coding
What is the medicine, and the reason of the usage?	ما هو الدواء و ما هي دواعي استعماله	SFDA(File#)_S0_(Sentence#)
Before taken the medicine	قبل القيام بتناول الدواء	SFDA(File#)_S1_(Sentence#)
How to take the medicine?	كيف تتناول الدواء	SFDA(File#)_S2_(Sentence#)
Possible side effects	الآثار الجانبية المحتملة	SFDA(File#)_S3_(Sentence#)
How to save the medicine?	كيف تقوم بحفظ الدواء	SFDA(File#)_S4_(Sentence#)
Important information about medicine ingredients.	معلومات مهمة حول بعض مكونات الدواء	SFDA(File#)_S5_(Sentence#)
KAAHE section (English)	**KAAHE section (Arabic)**	**Sentence coding**
Therapeutic Categories and drug dosage and	التَّصنيفُ العِلاجيُّ للدَّواء والجرعةُ الدَّوائيَّة	KAAHE(File#)_S0_(Sentence#)
mechanism of action	آلِيَّةَ عَمَل الدَّواء	KAAHE(File#)_S1_(Sentence#)
The reason for the medicine use	دَواعي استِعمال الدَّواء	KAAHE(File#)_S2_(Sentence#)
contraindication	مَوانعُ استِعمال الدَّواء	KAAHE(File#)_S3_(Sentence#)
Best way of taking medicine	ما هي الطَّريقةُ المُثلى لاستِعمال هذا الدَّواء؟	KAAHE(File#)_S4_(Sentence#)
Drug and foods interactions	تَداخلُ الدَّواء مع الطَّعام	KAAHE(File#)_S5_(Sentence#)
Medicine Interference with other medications	تَداخلُ الدَّواء مع الأدوية الأخرى	KAAHE(File#)_S6_(Sentence#)
What to do if you missed a dose?	ماذا أفعلُ إذا تأخَّرتُ عن موعد إحدى الجرعات؟	KAAHE(File#)_S7_(Sentence#)
Precautions that must be taken when using this medicine	ما هي الاحتياطاتُ التي يجب مُراعاتُها لدى استِعمال هذا الدَّواء؟	KAAHE(File#)_S8_(Sentence#)
What are the common side effects of this medicine?	ما هي التأثيراتُ الجانبيَّة الشَّائعة لهذا الدَّواء؟	KAAHE(File#)_S9_(Sentence#)
What you must monitor when using this medicine?	ماذا يجب على المَرء مراقبتُه عندَ استِعمال هذا الدَّواء؟	KAAHE(File#)_S10_(Sentence#)
What are the reasons for calling the health care resource "doctor"?	ما هي الأسبابُ التي تدعو لاستدعاء مورد الرعاية الصحِّيّة " الطَّبيب " على الفور؟	KAAHE(File#)_S11_(Sentence#)
What you supposed to follow when store this medicine?	ما المَفروضُ اتِّباعُه لدى تَخزين هذا الدَّواء؟	KAAHE(File#)_S12_(Sentence#)
General Instructions	إرشاداتٌ عامَّة	KAAHE(File#)_S13_(Sentence#)

Table 2 Words and sentences distribution

Source	SFDA	KAAHE	Total
Word count	31995	29400	61395
Sentences	2231	2245	4476

4. Annotation Methodology

We followed the annotation process pipeline defined by (Pustejovsky and Stubbs, 2012) to create an ARC-WMI with a readability annotations. Figure 2 shows the workflow and the main phases for the readability annotation process. The annotation modelling and guidelines define the annotation policy for the annotators and they identify the annotation values to be assigned for each sentence. Our readability guidelines focused on identifying the readability levels and they described how the difficulty level should be assigned for a given sentence. We defined three levels of text readability (Easy, Intermediate, and Difficult), as shown in Table 4.

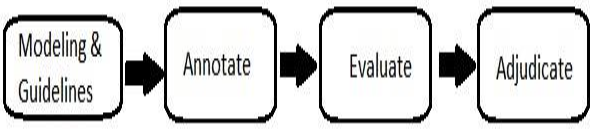

Figure 2 Annotation process pipeline

These levels were derived from a study by (Leroy et al., 2008), which evaluates the sentence based on the vocabulary used, the syntax structure, and the overall understanding. In the annotation phase, the sentence was judged for its readability by five health-care professionals' annotators. Each sentence was evaluated by two annotators to ensure the overlap between the annotation values.

In addition, these expert annotators have a pharmacy education background to ensure they can easily interpret and follow (Leory et al. ,2008) health information evaluation criteria. Each sentence was evaluated as "Easy" "Intermediate," or "Difficult," where readability was defined as a subjective judgment of how easily a reader could extract the information from the WMI.

In the evaluation phase, we measured the efficiency of the annotation's results using the Inter-Annotator Agreement (IAA) score. We calculated the IAA using kappa statistics for comparing two annotations against each other, based on Landis and Koch guidelines (Landis and Koch, 1977), to interpret the kappa value and define the agreement level.

Table 3 Example annotations from the corpus

Sentence ID	Sentence	English translation	annotator 1	annotator 2
KAAHE37_S0_3	أمَّا جرعةُ الصِّيانَة فهي 25-100 ملغ/اليوم على دفعة أو دفعتين بعدَ أسبوعين من بَدء العِلاج عندَ الضَّرورة؛ ويمكن إضافةُ المدرَّات حسب الحاجة.	The maintenance dose is 25-100 mg/ day on a batch or two batches after two weeks of starting treatment, it is possible to add diuretics when necessary	3	2
KAAHE36_S12_45	يُحفَظ الدَّواءُ في درجة حرارة الغرفة.	Keep the medicine at room temperature	1	1
KAAHE11_S3_11	إذا كان المَريضُ يعاني من أمراض الكبد أو من إسهال شَديد يُسمَّى التهاب القولون الغِشائي الكاذب.	If the patient is suffering from liver disease or from severe diarrhea called pseudo-colitis.	2	1

Table 4 Annotation guideline

Readability level	Definition
Easy	Contains small number of medical vocabulary and syntax structure used by the average consumer and he/she can understand the sentence without any help.
Intermediate	Contains medical vocabulary and syntax structure used in consumer health education and he/she can understand the sentence as consumer health education.
Difficult	Contains many medical vocabulary and syntax structure used by health professionals. Only health professionals can understand the sentence.

Table 5 shows the resulting IAA with average Inter-Annotator agreement 22% for the complete annotated dataset. This result indicates a fair agreement level with noticeable fluctuation in the agreement levels between the annotators. To resolve the conflict we used a third party judge to settle the differences in the annotation set. Table 3 presents sample of annotations values from dataset. Considering that the guideline definitions were derived from (Leory et al. 2008), still the annotators find it difficult to distinguish between Intermediate and Easy sentences. In addition the annotators tend to choose in case of uncertainty the intermediate level.

Adjudicating was conducted to resolve the conflicts between the annotators' results. In the annotation set, the differences between the annotations occurred because the text readability is based on the annotator's intuition to evaluate the difficulty level of the text. The differences between the annotations are expected and they are legitimate, based on the nature of the readability process (Finlayson, 2011). To finalize the results of the annotation, an adjudicator was employed to compare between the annotation values and to resolve any conflicts between the annotator's assigned values, to produce the final version of the annotated corpus. Table 6 shows the distribution of the sentences for each category, with an average sentence length of 15 for Easy, 21 for Intermediate, and 25 for Difficult.

Table 5 The resulted Inter-Annotator Agreement (IAA) score

Annotator Pair\agreement	IAA (Kappa)
Annotator (1 & 2)	0.48
Annotator (2 & 3)	0.11
Annotator (3 & 4)	0.028
Annotator (4 & 5)	0.28

Table 6 Distribution of sentences

	Easy	Intermediate	Difficult
Sentences count	3224	918	334
Words count (per sentence)	38501	15815	7079
AVG sentence length	15	21	25

5. Conclusion and Future Directions

In this paper, we presented the ARC-WMI Readability Corpus for WMIs, which is the first computationally analyzed Arabic corpus for readability assessment for the health domain. This corpus contains over 61k words and 4476 sentences annotated with three readability levels (Easy, Intermediate, and Difficult). We believe that a readability annotated corpus would be extremely valuable for future developments in computational readability research, especially for health literacy studies. Future work includes a further extension of the corpus along with guideline enhancement to improve the overall IAA results. The IAA values, can be improved by the experience gained over time by the annotators during the annotation process and by updating the annotation guidelines to simplify the readability assessment criteria and include a clear criteria for uncertainty cases as well. Finally, we will soon release the preliminary version of ARC-WMI corpus [3] under a Creative Commons License, so the research community can benefit from it.

Acknowledgment

This Project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (INF 2822).

References

Abdul-Mageed, M., Diab, M.T., 2012. AWATIF: A Multi-Genre Corpus for Modern Standard Arabic Subjectivity and Sentiment Analysis. Presented at the LREC, Citeseer, pp. 3907–3914.

Baker, J.P., 1997. Consistency and accuracy in correcting automatically tagged data. Garside et al.(1997) 243–250.

Basile, V., Bos, J., Evang, K., Venhuizen, N., 2012. Developing a large semantically annotated corpus. Presented at the LREC, pp. 3196–3200.

Bowman, S.R., Angeli, G., Potts, C., Manning, C.D., 2015. A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326.

Brants, S., Dipper, S., Hansen, S., Lezius, W., Smith, G., 2002. The TIGER treebank. Presented at the Proceedings of the workshop on treebanks and linguistic theories.

Deng, Y., Declerck, T., Lendvai, P., Denecke, K., 2016. The Generation of a Corpus for Clinical Sentiment Analysis. Presented at the International Semantic Web Conference, Springer, pp. 311–324.

Dukes, K., Habash, N., 2010. Morphological Annotation of Quranic Arabic. Presented at the LREC.

Finlayson, M.A., 2011. The Story Workbench: An Extensible Semi-Automatic Text Annotation Tool. Presented at the Intelligent Narrative Technologies.

Hewitt, M., 2012. Facilitating State Health Exchange Communication Through the Use of Health Literate Practices: Workshop Summary. National Academies Press.

Kandula, S., Zeng-Treitler, Q., 2008. Creating a gold standard for the readability measurement of health texts. Presented at the AMIA.

Koo, M., Krass, I., Aslani, P., 2006. Enhancing patient education about medicines: factors influencing reading and seeking of written medicine information. Health Expectations 9, 174–187.

Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. biometrics 159–174.

Leroy, G., Miller, T., Rosemblat, G., Browne, A., 2008. A balanced approach to health information evaluation: A vocabulary-based naïve Bayes classifier and readability formulas. Journal of the American Society for Information Science and Technology 59, 1409–1419.

Pakhomov, S.V., Coden, A., Chute, C.G., 2006. Developing a corpus of clinical notes manually annotated for

[3] https://github.com/iwan-rg/ARC-WMI

part-of-speech. International journal of medical informatics 75, 418–429.

Pustejovsky, J., Stubbs, A., 2012. Natural language annotation for machine learning. O'Reilly Media, Inc.

Rosemblat, G., Logan, R., Tse, T., Graham, L., 2006. Text features and readability: expert evaluation of consumer health text. Presented at the Mednet 2006: 11th World Congress on Internet in Medicine the Society for Internet in Medicine, Citeseer.

Van Oosten, P., Hoste, V., 2011. Readability annotation: Replacing the expert by the crowd. Presented at the Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, pp. 120–129.

Zaghouani, W., Awad, D., 2016. Building an Arabic Punctuated Corpus. Presented at the Qatar Foundation Annual Research Conference Proceedings, HBKU Press Qatar, p. SSHAPP3148.

ArSAS: An Arabic Speech-Act and Sentiment Corpus of Tweets

AbdelRahim A. Elmadany[1], **Hamdy Mubarak**[2], **Walid Magdy**[3]
[1]Jazan University, Saudi Arabia
[2]Qatar Computing Research Institute, HBKU, Qatar
[3]School of Informatics, Edinburgh University, UK
aelmadany@jazanu.edu.sa, hmubarak@hbku.edu.qa, wmagdy@inf.ed.ac.uk

Abstract

Speech acts are the type of communicative acts within a conversation. Speech act recognition (aka classification) has been an active research in recent years. However, much less attention was directed towards this task in Arabic due to the lack of resources for training an Arabic speech-act classifier. In this paper we present ArSAS , an Arabic corpus of tweets annotated for the tasks of speech-act recognition and sentiment analysis. A large set of 21k Arabic tweets covering multiple topics were collected, prepared and annotated for six different classes of speech-act labels, such as expression, assertion, and question. In addition, the same set of tweets were also annotated with four classes of sentiment. We aim to have this corpus promoting the research in both speech-act recognition and sentiment analysis tasks for Arabic language.

1. Introduction

Understanding user's Speech act within conversations is an important research task of Natural Language Understanding (NLU). Speech act task could be defined as identifying the intention of a speaker in producing a particular utterance of few words (e.g. sentence in a conversation or a tweet) (Webb, 2010), where the intention can be expression of a feeling, asking question, recommending something ... etc. Speech act recognition is becoming an essential task for many NLU applications such as summarization (McKeown et al., 2007), question answering (Hong and Davison, 2009), and chat-bots (Feng et al., 2006).

Speech act recognition is usually applied to conversations, such as dialogues and chatting platforms, which is mainly called synchronous conversations (Dhillon et al., 2004; Jurafsky, 1997). Other work focused on asynchronous conversations, where the discussion is to open audience and sequence of conversation is not fully synchronized; e.g. forums and social media (Tavafi et al., 2013; Oya and Carenini, 2014; Vosoughi and Roy, 2016b). Recently, social media platforms, such as Twitter, have become a major mean of communication between users. Consequently, understanding the speech act of user posts on these platforms became of increasing importance (Vosoughi and Roy, 2016b). The main objective of classifying speech act of a social post goes beyond the literal meaning of text, where it considers how the context and intention contribute to the meaning of the post (Vosoughi and Roy, 2016b). Most of the work on this task focused on English, while almost no attention was directed towards highly inflected languages such as Arabic.

In this paper, we present ArSAS, a manually annotated **Ar**abic **S**peech **A**ct and **S**entiment corpus of tweets. To our knowledge, ArSAS is considered the first corpus of Arabic speech act on Twitter. The corpus consists of a set of more than 21k Arabic tweets that are manually annotated for six differences classes of speech-act {Assertion, Expression, Recommendation, Respect, Question and Misc}. In addition, tweets are also annotated for four different categories of sentiment {Positive, Negative, Neutral and Mixed}, which is considered the largest Arabic tweets corpus that is labeled for four categories of sentiment. ArSAS dataset is publicly available for free for research purposes[1].

2. Background

2.1. Speech Act

One of the most essential steps in human-computer interaction systems, aka dialogue Systems, is understanding user's need. This process is called "language understating component", "Dialogue Acts" or "Speech Acts". Speech act recognition (also know as classification) task is labeling the speaker's intention in producing a particular utterance. The speech act terminology is approximately the equivalent of the speech act of (Searle, 1969), where it was presented as a fundamental concept of linguistic pragmatics analyzing; for example, what it means to ask a question or make a statement. Although major dialogue theories treat dialogue acts as a central notion, the conceptual granularity of the used speech act labels/classes varies considerably among alternative analyses, depending on the application or domain (Webb and Hardy, 2005). Within the field of computational linguistics, recent work closely linked to the development and deployment of spoken language dialogue systems has focused on some of the conversational roles such acts can perform. Most of the previous research on speech act is widely used with data transcribed from telephone or face-to-face conversations, which is also known as synchronous conversations (Dhillon et al., 2004; Jurafsky, 1997).
Recently, social media platforms, such as Twitter, became a hub for users to communicate and discuss various topics. These communicative acts among social media users are seen as a kind of asynchronous conversations, which can include spreading news, asking questions, or expressing feelings, which all fall under the scope of "speech act". Classifying speech act of social media posts can provide a new dimension to study social media content as well as providing real-life data to validate or reject claims in the

[1]http://homepages.inf.ed.ac.uk/wmagdy/resources.htm

speech act theory (Zhang et al., 2012). Speech act classification of tweets is considered fairly new task. Recent work mostly focused on classifying speech act of tweets on trending topics. (Zhang et al., 2012; Zhang et al., 2011) proposed a speech act classification method to understand Twitter users' behavior through a set of word-based and character-based features. (Nemer, 2015) proposed a system for understanding celebrity (e.g. Oprah Winfrey and Britney Spears) speech act on Twitter. They investigated celebrities' speech patterns on Twitter and whether they mostly talk to fans, and how they communicate with different audiences. (Vosoughi and Roy, 2015; Vosoughi and Roy, 2016a) proposed two systems based on assertion speech act detection. The assertion act is an utterance that commits the speaker to the truth of the expressed proposition. For example, the tweet "*there is a third bomber on the roof*" contains an assertion, while the tweet "*I hate reporters!*" contains an expression. They proved that assertion is important to identify rumors and track stories about real-world events. Moreover, they showed that more than half of tweets about events do not contain assertions. (Vosoughi and Roy, 2016b) proposed automatic speech act classifier for tweets based on semantics features such as opinion and vulgar words, emotions, speech act verb, n-grams, syntactic features, Twitter-specific Characters (# and @), abbreviations, and dependency Sub-trees. They examined four classifiers: Naive Bayes (NB), decision trees, logistic regression, and SVMs. All previous work mainly focused on speech act classification for English, while very limited work targeted speech act classification for Arabic.

2.2. Arabic Speech Act

To the best of our knowledge, there are two available corpora for Arabic speech act on synchronous conversations. First, TuDiCoI (Tunisian Dialect Corpus Interlocutor) consists of Railway information from the National Railway Company in Tunisia (SNCFT) which transcribed spoken Arabic dialogues and contains 12,182 utterances (Graja et al., 2013). Second, JANA corpus which is a multi-genre corpus of Arabic dialogues labeled for Arabic Dialogues Language Understanding (ADLU) at utterance level and comprising Spontaneous Speech Dialogues (SSD) and Instance Messages (IM) for Egyptian dialect. It contains 4,725 utterances and it is published on LDC (LDC2016T24) (Elmadany et al., 2016).

These two datasets were investigated in few research studies. (Graja et al., 2013) used the TuDiCoI corpus to develop a discriminative algorithm based on conditional random fields (CRF) to semantically label spoken Tunisian dialect turns which are not segmented into utterances. (Elmadany, 2016; Elmadany et al., 2018) utilized the JANA corpus to create a statistical dialogue analysis model for recognizing utterance's dialogue acts using a machine learning approach based on multi-classes hierarchical structure.

In addition, there are other few initiatives that studied Arabic speech acts classification, but on a much smaller scale using hand-crafted small dataset. (Shala et al., 2010) applied speech act classification for Arabic discourse using NB and decision trees classifiers on a dataset of about 400 utterances only collected from newspapers. (Bahou et al., 2008) proposed a method for the semantic representations of utterances of spontaneous Arabic speech based on the frame grammar formalism and tested on about 1,000 Tunisian national railway queries collected using Wizard-of-Oz technology. Another work (Lhioui et al., 2013) used the same Wizard-of-Oz technology but to collect a smaller set of 140 utterances only recorded from 10 speakers.

Previous work shows the huge limitation in the availability of annotated Arabic data for the task of speech act recognition. We believe that ArSAS would be the first stranded corpus for Arabic speech act classification for asynchronous conversations, which contains over 21k tweets labeled with fine-grained set of six different speech act classes.

2.3. Arabic Sentiment Analysis

Unlike speech act, there was some attention to Arabic sentiment analysis including few initiative to create standard corpora and lexicons for this task.

Early work on Arabic sentiment analysis focused on Modern Standard Arabic (MSA) (Abbasi et al., 2008; Abdul-Mageed et al., 2011). Later on, many initiatives started to focus on dialectal Arabic on social media (Mourad and Darwish, 2013; Abdul-Mageed et al., 2014). One of the initial work on sentiment analysis for Arabic tweets was presented by (Mourad and Darwish, 2013). They proposed expandable ArabSinti lexicon for both Modern Standard Arabic (MSA) news articles and dialectal Arabic tweets. They used 2,300 Arabic tweets annotated with five possible labels: neutral, positive, negative, both, or sarcastic. Another work by (Badaro et al., 2015) introduced a large-scale Standard Arabic sentiment lexicon (ArSenL) developed using a combination of English SentiWordnet (ESWN), Arabic WordNet, and the Arabic Morphological Analyzer (AraMorph). They developed a set of 28,760 words, but mainly in MSA. (Ibrahim et al., 2015) proposed a corpus of MSA and Egyptian dialect. The corpus is extracted from tweets, comments on hotel reservations and TV programs and product reviews annotated at the sentence level. It consists of 2,154 positive, 1,648 negative and 1,98 neutral texts. (Refaee and Rieser, 2014) proposed a corpus of Arabic tweets annotated for subjectivity and sentiment analysis consists of 6,894 tweets and annotated with four sentiment labels: positive, negative, neutral and mixed. More recent work in SemEval 2016 on a sentiment analysis task for multiple languages including Arabic (Kiritchenko et al., 2016) introduced a small dataset of 1,366 tweets. Another SemEval task for sentiment analysis for Arabic tweets was introduced in 2017 with a larger set of 9,455 Arabic tweets annotated with 3 labels: positive, negative, and neutral (Rosenthal et al., 2017). Another available copora on Arabic sentiment analysis was introduced by (Nabil et al., 2015), where they introduced the Arabic Sentiment Tweets Dataset (ASTD) which contains 10k Egyptian tweets annotated with four sentiment labels. Finally, (Al-Twairesh et al., 2017) developed a larger corpus that consists of 17k annotated tweets with the same four sentiment labels.

Our ArSAS corpus should be the current largest corpus for Arabic sentiment analysis with over 21k annotated tweets annotated with 4 different labels of sentiment.

Type	Arabic Topic	Translation
Events	الانتخابات الرئاسية المصرية	Egyptian presidential election
	تفجيرات سيناء	Sinai bombings
	حادث الواحات	Wahat attack
	تصفيات كأس العالم	World cup qualifications
	محاربة الفساد في السعودية	Fighting corruption in KSA
	منتدى شباب العالم	World youth forum
	اعتقال سلمان العودة	Arresting Salman Aloda
Entities	عبد الفتاح السيسي	Abdelfattah Al-Sisi
	المحامي خالد علي	Khaled Ali
	حمدين صباحي	Hamdeen Sabahi
	محمد صلاح	Mohamed Salah
	مرتضى منصور	Mortada Mansour
	ولي العهد السعودي	KSA crown prince
	الحوثيون	Houthis
	خليفة حفتر	Khalifa Haftar
Long Standing	الربيع العربي	Arab spring
	تيران وصنافير	Tiran and Sanafir
	ثورة يناير	January Revolution
	أزمة الخليج	Gulf crisis
	حصار ومقاطعة قطر	Qatar siege and boycott

Table 1: List of topics used to collect tweets

3. Corpus Creation

3.1. Data Collection

We used the Twitter API[2] to collect tweets on a set of topics we developed. We used (Zhao and Jiang, 2011; Vosoughi and Roy, 2016b) definitions for three different types of topic. A topic is an essay or article which discussed in one or more tweets. A type is the characteristic of topics, and is classified into:

- Long-Standing: Topics about articles that are commonly discussed over long period of time.
- *Entity*: Topics about celebrities or organizations.
- Event: Topics about an important thing that is happening.

We created a set of 20 topics of the three types above which covers controversial topics that potentially get discussion on social media, which would be highly suitable for both the tasks of speech act recognition and sentiment analysis. Table 1 shows the list of topics we developed and used to collect the tweets.

We collected a set of 62,690 tweets in the period 1-15 November 2017. We applied some data filtering by removing short tweets that contain fewer than three words (without counting hashtags, user mentions, and URLs). Then we randomly selected a set of 21,064 tweets for annotation, where 6151, 6146, 8767 tweets were covering the long-standing, entity, and event topics respectively.

3.2. Labels Schema

Each tweet in our collection prepared for annotation with two labels for speech act and sentiment. We used a list of six speech act tags based on Searleś speech act taxonomies (Searle, 1969; Searle, 1975) as follows:

[2]https://dev.twitter.com/

Arabic Tweet	Speech act	Sentiment
الرياضة تنتخب - شريف عبد القادر : الزمالك في أزمة ولا أحد ينكر طفرة مرتضى منصور داخل النادي	Assertion	Mixed
الكرة الايطالية تحتاج لتركي ال الشيخ	Recommendation	Neutral
الربيع العربي رغم الالتفاف عليه الا أنه إشعاع من الحرية	Expression	Positive
ليه محمد صلاح؟ ☺ ☺	Question	Positive
اطالب الرئيس ببناء مدينة لها سور عالى فى اقاصى الصحراء وليتم حجز كل المعاتيه الذين امنو بما اسموها ثورة يناير	Request	Negative
سأرفع الحد الأدنى للأجور إلى 2000 جنيه وسأنفذ حكم تيران وصنافير - وتعهد أيضا بالإفراج عن المحبوسين	Miscellaneous	Neutral

Table 2: Samples of annotated tweets with speech act and sentiment

1. Assertion: user declares some proposition such as stating, claiming, reporting, or announcing.
2. Recommendation: user recommends something.
3. Expression: user expresses some psychological state such as thanking, apologizing, or congratulating.
4. Question: user asks a question such as why, what, or confirmation.
5. Request: user asks for something such as ordering, requesting, demanding, or begging.
6. Miscellaneous: user committed to some future action such as promising or offering.

For sentiment labels, we used the standard four sentiment tags: positive, negative, mixed (contains both positive and negative sentiment), or neutral (no opinion or sentiment disclosed).

Table 2 shows illustrative examples of each of the speech act and sentiment tags.

3.3. Data Annotation

For tweets annotation, we created a job on CrowdFlower crowdsourcing platform where we showed tweets to annotators and asked them to classify speech act and sentiment for each tweet into one of the above-described tags. Guidelines and examples of tweets for each tag were presented to annotators for better understanding. We restricted annotation to workers from all Arab countries who have "Arabic" language in their profile. Each tweet was judged by at least three annotators.

Quality of annotation was controlled by utilizing 70 hidden test questions; each has the correct answer(s) for both speech act and sentiment. These test questions were selected such that their answers, as selected by two language experts, are matched. Annotators on CrowdFlower were required to get at least 70% of the hidden test questions correctly to continue. Otherwise, they get excluded from the job and their work gets discarded. Around 500 annotators participated successfully in the annotation process which gives the diversity of opinions needed for such tasks.

Agreement among annotators was 87% for speech act, and 79% for sentiment, which indicates that annotation

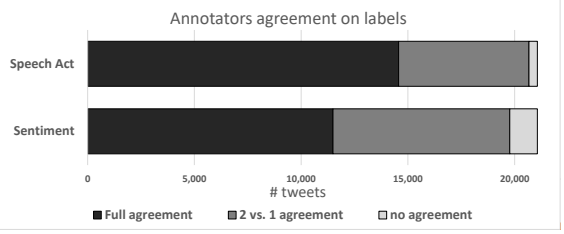

Figure 1: The number of tweets that have full, partial, or no agreement for each of the speech act and sentiment labels

of speech act might be simpler and more straightforward than the arguable sentiment. However, both inter-annotator agreements are considered high, especially for an annotation task with three annotators and 4+ choices.

Crowdflower provides a confidence score with each annotated tweet that represents the confidence in the quality of the label. For a three annotators per tweet setup, the confidence score would range between 0.3 and 1 according to two factors: 1) annotator quality level; and 2) agreement among annotators. A confidence score of 1 means that the three annotators selected the same label. A score around 0.6 refers to two annotators agreeing on a given label while the third selecting another one. A score around 0.3 refers to having the three annotators selecting three different labels, and in this case the label selected by the annotator with the highest quality value is considered. Figure 1 shows the number of tweets that have full, partial, or no agreement for each of the speech act and sentiment labels. As shown, the majority of the tweets got full agreement for both tasks (69% and 55% for speech act and sentiment labels respectively). The number of tweets that received different label from each annotator is very small (2% and 6% of all tweets for speech act and sentiment labels respectively). We could exclude these tweets, however we preferred to keep in our collection as an example of challenging tweets.

4. ArSAS Corpus Characteristics

ArSASreleased dataset contains the following information:

- `ID`: ID of the tweet.
- `Text`: the original unprocessed text of the tweet
- `topic`: topic type of the keyword used to collect the tweet.
- `Sentiment`: selected sentiment label.
- `Sentiment Conf.`: Confidence score of sentiment label.
- `Speech Act`: selected speech act label.
- `Speech Act Conf.`: Confidence score of the speech act label.

Figure 2(a) shows the distribution of speech act labels in our ArSAS corpus after annotation. As shown, the majority of the tweets are labeled as either Expression (56%) or Assertion (39%), and the remaining labels are used in only 5% of the cases. This highly unbalanced distribution is similar to the English tweets corpus used in (Vosoughi and Roy, 2016b).

Figure 2(b) shows the distribution of sentiment labels. Tweets having negative sentiment represent one third of the tweets, while those with positive sentiment represent around quarter of the tweets, and one third of the tweets have no (neutral) sentiment. Only 6% of the tweets have mixed sentiment.

Table 3 shows the fine-grained distribution of speech act tags by topics type and sentiment tags. As can be seen in the table, the majority of the Assertion tweets are coming from the 'Events' topics, while the tweets with other speech acts have less bias towards the topics. Also, it can be noticed that the largest two speech act tags, Assertion and Expression, have very different distributions for the sentiment, where most of the Assertion tweets have no sentiment (neutral), while most of the tweets with Expression speech act have polarized sentiment, most of them are negative. These observations show the value of having a corpus labeled for both speech act and sentiment, since one of the two tasks can be used as an effective feature to predict the other.

5. Conclusion

In this paper, we introduce ArSAS, a large dataset of Arabic tweets annotated for both speech acts and sentiment. ArSAS consists of 21k Arabic tweets written in multiple Arabic dialects as observed by examining different samples. The tweets in the corpus were extracted and collected using 20 controversial topics in different countries that are expected to have hot discussions among Twitter users. The tweets collection did not rely on emotions or sentiment keywords to avoid data bias to a given lexicon, especially for the task of sentiment analysis. The corpus is annotated with six speech act labels and four sentiment labels. The annotation process was applied using a crowdsourcing platform by having at least three annotators labeling each tweet. An inter-annotator agreement of 87% and 79% was achieved for the speech act and sentiment labels respectively. To the best of our knowledge, ArSAS is the largest annotated corpus of speech act and sentiment in Arabic. In addition, it is considered the first Arabic corpus annotated for the speech act recognition in tweets. We hope that our corpus would bring the attention to the speech act recognition task for Arabic and further promote the research in Arabic sentiment analysis. Moreover, it can be applied for applications that combines both tasks.

ArSAS corpus is freely available online as an open-source for researchers interested in Arabic speech act and sentiment analysis and could be downloaded from `http://homepages.inf.ed.ac.uk/wmagdy/resources.htm`.

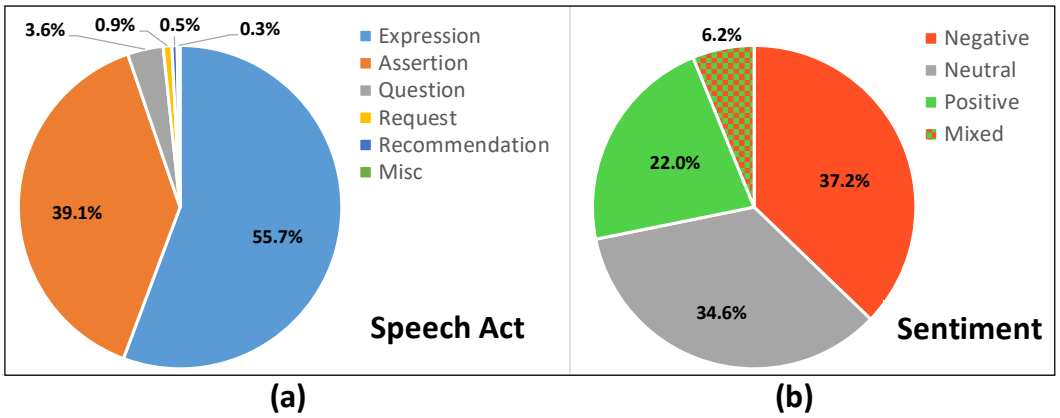

Figure 2: The distribution of the speech act tags (a) and sentiment tags (b) in the ArSAS corpus

	Topics Types			Sentiment Analysis tags			
Speech Act (# tweets)	**Long-standing**	**Entities**	**Events**	**Neutral**	**Positive**	**Negative**	**Mixed**
Assertion (8,221)	627	2,097	5,497	6,666	962	488	105
Recommendation (107)	18	55	34	23	36	36	12
Question (751)	327	219	205	248	47	403	53
Request (180)	23	94	63	30	63	66	21
Expression (11,745)	5,126	3,658	2,961	289	3,514	6,835	1,107
Miscellaneous (60)	30	23	7	23	21	12	4

Table 3: The distribution of speech act tags via topics type and sentiment analysis tags

6. References

Abbasi, A., Chen, H., and Salem, A. (2008). Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12.

Abdul-Mageed, M., Diab, M. T., and Korayem, M. (2011). Subjectivity and sentiment analysis of modern standard arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 587–591. Association for Computational Linguistics.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Al-Twairesh, N., Al-Khalifa, H., Al-Salman, A., and Al-Ohali, Y. (2017). Arasenti-tweet: A corpus for arabic sentiment analysis of saudi tweets. *Procedia Computer Science*, 117:63–72.

Badaro, G., Baly, R., Akel, R., Fayad, L., Khairallah, J., Hajj, H., Shaban, K., and El-Hajj, W. (2015). A light lexicon-based mobile application for sentiment mining of arabic tweets. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 18–25.

Bahou, Y., Belguith, L. H., and Hamadou, A. B. (2008). Towards a human-machine spoken dialogue in arabic. In *6th Language Resources and Evaluation Conference (LREC 2008), Workshop HLT Within the Arabic World. Arabic Language and Local Languages Processing Status Updates and Prospects, Marrakech, Morocco*.

Dhillon, R., Bhagat, S., Carvey, H., and Shriberg, E. (2004). Meeting recorder project: Dialog act labeling guide. Technical report, INTERNATIONAL COMPUTER SCIENCE INST BERKELEY CA.

Elmadany, A., Abdou, S., and Gheith, M. (2016). Jana: A human-human dialogues corpus for egyptian dialect (ldc2016t24).

Elmadany, A., Abdou, S., and Gheith, M. (2018). Improving dialogue act classification for spontaneous arabic speech and instant messages at utterance level. In *11th edition of the Language Resources and Evaluation Conference*.

Elmadany, A. (2016). *Automatic Act Classification for Arabic Dialogue Context*. Thesis.

Feng, D., Shaw, E., Kim, J., and Hovy, E. (2006). An intelligent discussion-bot for answering student queries in threaded discussions. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 171–177. ACM.

Graja, M., Jaoua, M., and Belguith, L. H. (2013). Discriminative framework for spoken tunisian dialect understanding. In *International Conference on Statistical Language and Speech Processing*, pages 102–110. Springer.

Hong, L. and Davison, B. D. (2009). A classification-based approach to question answering in discussion boards. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 171–178. ACM.

Ibrahim, H. S., Abdou, S. M., and Gheith, M. (2015). Mika: A tagged corpus for modern standard arabic and colloquial sentiment analysis. In *Recent Trends in Infor-*

mation Systems (ReTIS), 2015 IEEE 2nd International Conference on, pages 353–358. IEEE.

Jurafsky, D. (1997). Switchboard swbd-damsl shallow-discourse-function annotation coders manual. *www. dcs. shef. ac. uk/nlp/amities/files/bib/ics-tr-97-02. pdf*.

Kiritchenko, S., Mohammad, S., and Salameh, M. (2016). Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In *SemEval@ NAACL-HLT*, pages 42–51.

Lhioui, C., Zouaghi, A., and Zrigui, M. (2013). A combined method based on stochastic and linguistic paradigm for the understanding of arabic spontaneous utterances. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 549–558. Springer.

McKeown, K., Shrestha, L., and Rambow, O. (2007). Using question-answer pairs in extractive summarization of email conversations. *Computational Linguistics and Intelligent Text Processing*, pages 542–550.

Mourad, A. and Darwish, K. (2013). Subjectivity and sentiment analysis of modern standard arabic and arabic microblogs. In *WASSA@ NAACL-HLT*, pages 55–64.

Nabil, M., Aly, M. A., and Atiya, A. F. (2015). Astd: Arabic sentiment tweets dataset. In *EMNLP*, pages 2515–2519.

Nemer, D. (2015). Celebrities acting up: A speech act analysis in tweets of famous people. *Social Networking*, 5(01):1.

Oya, T. and Carenini, G. (2014). Extractive summarization and dialogue act modeling on email threads: An integrated probabilistic approach. In *SIGDIAL Conference*, pages 133–140.

Refaee, E. and Rieser, V. (2014). An arabic twitter corpus for subjectivity and sentiment analysis. In *LREC*, pages 2268–2273.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Searle, J. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press.

Searle, J. (1975). A taxonomy of illocutionary acts. *Language, Mind, and Knowledge: Minneapolis Studies in the Philosophy of Science*, 7:344–369.

Shala, L., Rus, V., and Graesser, A. C. (2010). Automated speech act classification in arabic. *Subjetividad y Procesos Cognitivos*, 14:284–292.

Tavafi, M., Mehdad, Y., Joty, S. R., Carenini, G., and Ng, R. T. (2013). Dialogue act recognition in synchronous and asynchronous conversations. In *SIGDIAL Conference*, pages 117–121.

Vosoughi, S. and Roy, D. (2015). A human-machine collaborative system for identifying rumors on twitter. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 47–50. IEEE.

Vosoughi, S. and Roy, D. (2016a). A semi-automatic method for efficient detection of stories on social media. In *ICWSM*, pages 707–710.

Vosoughi, S. and Roy, D. (2016b). Tweet acts: A speech act classifier for twitter. In *The 10th International Aaai Conference On Web And Social Media (ICWSM-16)*, pages 711–715.

Webb, N. and Hardy, H. (2005). Data-driven language understanding for spoken language dialogue. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*, pages 23–29.

Webb, N. (2010). *Cue-based dialogue act classification*. Ph.D. thesis, University of Sheffield, Department of Computer Science.

Zhang, R., Gao, D., and Li, W. (2011). What are tweeters doing: Recognizing speech acts in twitter. *Analyzing Microtext*, 11:05.

Zhang, R., Gao, D., and Li, W. (2012). Towards scalable speech act recognition in twitter: tackling insufficient training data. In *Proceedings of the Workshop on Semantic Analysis in Social Media*, pages 18–27. Association for Computational Linguistics.

Zhao, X. and Jiang, J. (2011). An empirical comparison of topics in twitter and traditional media. *Singapore Management University School of Information Systems Technical paper series*.

ArSEL: A Large Scale Arabic Sentiment and Emotion Lexicon

Gilbert Badaro, Hussein Jundi, Hazem Hajj, Wassim El-Hajj, Nizar Habash[†]
American University of Beirut, Beirut, Lebanon
[†]New York University Abu Dhabi, Abu Dhabi, UAE
ggb05@aub.edu.lb, haj14@mail.aub.edu, {hh63,we07}@aub.edu.lb, nizar.habash@nyu.edu

Abstract

With the advancement of Web 2.0, social networks experienced a great increase in the number of active users reaching 2 billion active users on Facebook at the end of 2017. Consequently, the size of text data on the Internet increased tremendously. This textual data is rich in knowledge, which attracted many data scientists as well as computational linguists to develop resources and models to automatically process the data and extract useful information. One major interest is sentiment and emotion classification from text. In fact, learning the opinion and emotions of people is important for businesses, marketers, government, politicians, etc. While focus had been given to sentiment analysis, recently emotion analysis has captured great interest as well. Several resources were developed for emotion analysis from text for English, however, very few targeted Arabic text. We present in this paper, ArSEL, the first large scale **Ar**abic **S**entiment and **E**motion **L**exicon. ArSEL is built in a way to augment the publicly available Arabic Sentiment Lexicon, ArSenL, and to generate a large scale lexicon that includes emotion and sentiment labels for almost every lemma in ArSenL. We also show the efficiency of using ArSEL in emotion regression and classification tasks using an Arabic translated version of annotated data from SemEval 2007 "Affective Task" as well as SemEval 2018 Task1 "Affect in Tweets" Arabic dataset. Coverages of 91% and 84% are achieved on the two datasets respectively. An improvement of 30% compared to majority baseline is achieved in terms of average F1 measure for emotion classification on SemEval 2018 Arabic dataset. ArSEL is publicly available on `http://oma-project.com`.

Keywords: Emotion Lexicon, Arabic Natural Language Processing, Emotion Classification, Regression

1. Introduction

The task of emotion recognition has been extensively studied from different modalities. For instance, several researchers tried to predict users' emotion by looking at their interaction with computers (Cowie et al., 2001; Pantic and Rothkrantz, 2003; Brave and Nass, 2003; Fragopanagos and Taylor, 2005; Jaimes and Sebe, 2007; Hibbeln et al., 2017; Patwardhan and Knapp, 2017; Constantine et al., 2016). Others have tried to assign to facial expressions emotion labels (Busso et al., 2004; Goldman and Sripada, 2005; Gunes and Piccardi, 2007; Trad et al., 2012; Wegrzyn et al., 2017). Recently, with the increase of textual data on the Web, computational linguists and data scientists started looking at emotion analysis from text. In fact, recognizing emotions of users is critical for different applications: first, it helps businesses and companies sense the feedback of its clients expressed on the Internet and consequently adapt their marketing strategies (Bougie et al., 2003); second, it allows providing customers with better personalized recommendations whether for advertisements or products (Mohammad and Yang, 2011) on top of collaborative filtering based recommender systems (Badaro et al., 2013; Badaro et al., 2014c; Badaro et al., 2014d); third, it can help in tracking emotions of users towards politicians, movies, music, products, etc, (Pang et al., 2008); fourth, it allows developing complex search algorithms that provide advanced search features filtered by emotions (Knautz et al., 2010) and last but not least, it allows a more accurate prediction of stock market prices (Bollen et al., 2011).

Some efforts have already been placed in developing emotion classification models from text (Shaheen et al., 2014; Houjeij et al., 2012; Abdul-Mageed and Ungar, 2017; Felbo et al., 2017). Since sentiment lexicons helped in improving the accuracy of sentiment classification models (Liu and Zhang, 2012; Taboada et al., 2011), several researchers are working on developing emotion lexicons for different languages such as English, French and Chinese (Mohammad, 2017; Bandhakavi et al., 2017; Yang et al., 2007; Poria et al., 2012; Mohammad and Turney, 2013; Das et al., 2012; Mohammad et al., 2013; Abdaoui et al., 2017; Staiano and Guerini, 2014). While sentiment is usually represented by three labels namely positive, negative or neutral, several representation models exist for emotions such as Ekman representation (Ekman, 1992) or Plutchik model (Plutchik, 1980; Plutchik, 1994) that includes Ekman's six emotions in addition to two labels: trust and anticipation. Despite the efforts for creating large scale emotion lexicons for English, the size of existing lexicons remain much smaller compared to sentiment lexicons. For example, DepecheMood (Staiano and Guerini, 2014), one of the largest publicly available emotion lexicon for English, includes around 37K while SentiWordNet (SWN) (Esuli and Sebastiani, 2007; Baccianella et al., 2010), a large scale English sentiment lexicon semi-automatically generated using English WordNet (EWN) (Fellbaum, 1998), includes around 150K terms annotated with three sentiment scores: positive, negative and objective. While some efforts have already been placed for developing emotion lexicons for English, we were only able to find two attempts for Arabic where the first emotion lexicon is a Google translation of an English Emotion lexicon, Emolex (Mohammad and Turney, 2013; Mohammad et al., 2013) and the second one is extracted from manually annotated Arabic documents for emotions (El Gohary et al., 2013). In fact, more work can be found related to sentiment analysis classification models for Arabic such as the work in (Badaro et al., 2014b; Badaro et al., 2015; Al Sallab et al., 2015;

Al-Sallab et al., 2017; Baly et al., 2017b; Abdul-Mageed, 2017) and to Arabic sentiment lexicon developments such as ArSenL (Badaro et al., 2014a), SIFAAT (Abdul-Mageed and Diab, 2012) and SANA (Abdul-Mageed and Diab, 2014). Developing emotion and sentiment classification models for Arabic is important given the tremendous increase of Arabic speaking users of Web 2.0. For instance, more than 11 million users are active on Twitter within the 22 Arab countries and more than 27 million tweets are generated daily.[1] Moreover, analyzing Arabic Twitter is a more complex task than MSA given that it includes different dialects with different characteristics (Baly et al., 2017a). Since the usage of sentiment lexicons in sentiment classification models showed significant improvement in the accuracy of such models (Al-Sallab et al., 2017), it is necessary to develop Arabic emotion lexicons for improved emotion classification models.

In this paper, we present ArSEL, the first publicly available large scale Arabic sentiment and emotion lexicon. ArSEL is an extension of ArSenL, where almost each lemma[2] in ArSenL is amended by eight emotion scores corresponding to: *afraid, amused, angry, annoyed, don't care, happy, inspired* and *sad*. The emotion scores are automatically obtained from DepecheMood (Staiano and Guerini, 2014), one of the largest publicly available English emotion lexicon. We first align DepecheMood with English WordNet (Fellbaum, 1998) and then, using synonymy semantic relation, we expand the coverage of DepecheMood and obtain EWN synsets annotated with emotion scores. Since ArSenL is linked to EWN 3.0, we can automatically assign the synsets' emotion scores to ArSenL lemmas. ArSEL can be used for several NLP tasks such as sentiment analysis, emotion analysis, or other semantic extraction tasks. It would be in particular useful for cases where it is desired to simultaneously extract the sentiment and emotion scores for words. In order to test the efficiency of ArSEL, we utilize ArSEL in emotion regression and classification tasks using unsupervised techniques similar to the way the efficiency of DepecheMood was tested with SemEval 2007 Affective Task dataset (Strapparava and Mihalcea, 2007). We also test the usefulness of ArSEL on a native Arabic dataset from SemEval 2018 Task1 "Affect in Tweets".[3]

The paper is organized as follows: in section 2, we present a literature review about emotion lexicon development. In section 3, we describe the approach followed for constructing ArSEL. In section 4, we evaluate ArSEL in emotion regression and classification tasks using first, SemEval 2007 news headlines data translated from English to Arabic using Google translate and second, SemEval 2018 Arabic Affect Tweets. We conclude the results of the paper in section 5 and present some ideas for future work.

[1]https://weedoo.tech/twitter-arab-world-statistics-feb-2017/

[2]For more information on issues of Arabic morphology in natural language processing, see (Habash, 2010).

[3]https://competitions.codalab.org/competitions/17751

2. Literature Review

We conduct a literature review on existing emotion lexicons for multiple languages. We present the techniques used to build the lexicons and the methods employed for evaluating their efficiency in emotion recognition tasks.

Strapparava et al. (2004) developed WordNet Affect by tagging specific synsets with affective meanings in EWN. They identified first a core number of synsets that represent emotions of a lexical database. They expanded then the coverage of the lexicon by checking semantically related synsets compared to the core set. They were able to annotate 2,874 synsets and 4,787 words. WordNet Affect was also tested in different applications such as affective text sensing systems and computational humor. WordNet Affect is of good quality given that it was manually created and validated, however, it is of limited size.

Mohammad and Turney (2013) presented challenges that researchers face for developing emotion lexicons and devised an annotation strategy to create a good quality and inexpensive emotion lexicon, EmoLex, by utilizing crowdsourcing. To create EmoLex, the authors first identified target terms for annotation extracted from Macquarie Thesaurus (Bernard and Bernard, 1986), WordNet Affect and the General Inquirer (Stone et al., 1966). Then, they launched the annotation task on Amazon's Mechanical Turk. EmoLex has around 10K terms annotated for emotions as well as for sentiment polarities. They evaluated the annotation quality using different techniques such as computing inter-annotator agreement and comparing a subsample of EmoLex with existing gold data. Moreover, they utilized Google translate to perform word translations into multiple languages including Arabic (Mohammad et al., 2013). However, the translation may include several errors: first, the translation may be incorrect since it is a word to word translation and second, the translation may be a transliteration instead in case the word is seen for the first time by the machine translator. Furthermore, the terms in the lexicon are not in their lemma form which make the lexicon harder to be utilized in an emotion classification task.

AffectNet (Cambria et al., 2012), part of the SenticNet project, includes also around 10K terms extracted from ConceptNet (Liu and Singh, 2004) and aligned with WordNet Affect. They extended WordNet Affect using the concepts in ConceptNet. While WordNet Affect, EmoLex and AffectNet include terms with emotion labels, Affect database (Neviarouskaya et al., 2007) and DepecheMood (Staiano and Guerini, 2014) include words that have emotion scores instead. Affect database extends SentiFul (Neviarouskaya et al., 2011) and covers around 2.5K words presented in their lemma form along with the corresponding part of speech tag.

DepecheMood is automatically built by harvesting social media data that were implicitly annotated with emotions. They utilize news articles from rappler.com. The articles are accompanied by Rappler's Mood Meter, which allows

readers to express their emotions about the article they are reading. DepecheMood includes around 37K lemmas along with their part of speech (POS) tags and the lemmas are aligned with EWN. Staiano and Guerini also evaluated DepecheMood in emotion regression and classification tasks in unsupervised settings. They claim that, although they utilized a naïve unsupervised model, they were able to outperform existing lexicons when tested on SemEval 2007 dataset (Strapparava and Mihalcea, 2007).

Bandhakavi et al. worked on constructing emotion lexicons using Tweets annotated with emotion labels (Bandhakavi et al., 2014; Bandhakavi et al., 2017). They experiment different techniques for lexicon generation: term frequency models and iterative models including generative and expectation maximization algorithms. Bandhakavi et al. evaluated the different lexicons on a Twitter dataset (Wang et al., 2012) and utilized a feature based supervised approach for classifying emotion.

While the above emotion lexicons were mainly developed for English, Yang et al. (2007) constructed an emotion lexicon for Chinese language. The authors used web blog corpora in order to extract the lexicon terms and assigned emotion scores using point wise mutual information measure. They created two different lexicons by varying the number of documents downloaded from the Web. They also evaluated the lexicons in an emotion classification task using different prediction methods.

Xu et al. (2010) also worked on constructing emotion lexicon for Chinese using graph-based algorithm which ranks words according to a few seed emotion words. The graph algorithm utilizes different similarity measures derived from dictionaries, unlabeled corpora and heuristic rules. In order to improve the quality of the lexicon, they mixed manual verification with the automatic assignment of emotions.

Abdaoui et al. (2017) presented Feel, an emotion and sentiment lexicon for French. Abdaoui et al. utilized NRC emotion lexicon (Mohammad et al., 2013) and translated its terms to French using multiple online translators. Then, a professional human translator validated the translation along with their emotion labels. Abdaoui et al. also claimed that FEEL outperformed other French emotion lexicons in emotion classification from texts.

In summary, several techniques are employed for building emotion lexicons and can be mainly grouped into two categories: the first one is based on manual annotation provided by professional individuals or through crowdsourcing, the second technique is rather automatic and lexicons are derived from annotated corpora. Only couple of papers worked on developing emotion lexicon for Arabic, thus, we focus on developing a large-scale Arabic emotion lexicon. We present next the methodology followed to construct automatically ArSEL by utilizing DepecheMood, EWN and ArSenL.

3. ArSEL

We describe in this section the process followed to construct ArSEL. We first briefly describe the harvested resources. Then, we present the expansion technique of DepecheMood and how we link it to ArSenL.

3.1. Resources

We make use of three resources: DepecheMood, English WordNet and ArSenL.

DepecheMood: (Staiano and Guerini, 2014) an emotion lexicon for English consisting of 37,771 words aligned with English WordNet. Each word along with its corresponding part of speech tag is annotated with 8 emotion scores (afraid, amused, angry, annoyed, don't care, happy, inspired and sad) derived automatically from annotated corpora collected from Rappler.com news website. Three variations of the lexicon were presented where the differences are related to the method of normalizing the emotion scores.

English WordNet 3.0: (Fellbaum, 1998; Fellbaum, 2010) is a hierarchical dictionary including more than 117,000 synsets and around 150,000 terms distributed among four part of speech tags: noun, verb, adjective and adverb. EWN has been used extensively in multiple natural language processing tasks and also for developing sentiment lexicons such as SentiWordNet (Esuli and Sebastiani, 2007; Baccianella et al., 2010) and emotion lexicons such as WordNet Affect (Strapparava et al., 2004).

ArSenL: (Badaro et al., 2014a) is a free publicly available large-scale Arabic Sentiment lexicon. ArSenL consists of Arabic lemmas assigned to EWN synsets along with three sentiment scores derived from English SentiWordNet. ArSenL was automatically developed by taking the union of two sentiment lexicons: the first one maps Arabic WordNet 2.0 (Black et al., 2006) to English SentiWordNet by using WordNet sense map files across WordNet versions 2.0, 2.1 and 3.0. The second lexicon is the result of performing gloss matching between English gloss terms of an Arabic lexical resource, SAMA (Standard Arabic Morphological Analyzer) (Graff et al., 2009), and EWN synset terms. In both sub-lexicons, the sentiment scores are obtained from English SentiWordNet. ArSenL includes 153,638 Arabic lemma-EWN synset pairs corresponding to 33,995 Arabic lemmas/POS tags annotated with three sentiment scores: positive, negative and objective.

We choose DepecheMood since it is the largest publicly available emotion lexicon in English and its terms are aligned with English WordNet. We benefit from the available alignment with English WordNet to expand the coverage of DepecheMood and obtain emotion scores for EWN synsets, in addition to emotion scores for an expanded list of EWN terms compared to those already in DepecheMood. We also utilize the advantage that ArSenL is connected to EWN synsets and hence, we automatically

assign emotion scores of EmoWordNet to corresponding ArSenL entries.

3.2. Expansion of DepecheMood and Link to ArSenL

In Figure 1, we show an overview of the steps followed to expand DepecheMood into EmoWordNet (steps grouped under DepecheMood Expansion) and then linking EmoWordNet to ArSenL to obtain ArSEL.

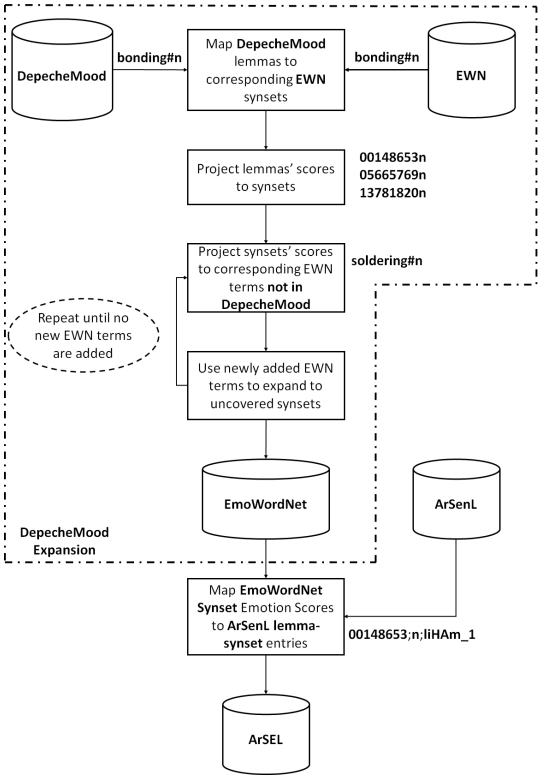

Figure 1: Overview of ArSEL Construction Methodology.

We detail first the steps utilized for expanding DepecheMood iteratively into what we name EmoWordNet.

Step 1: EWN synsets that include lemmas of DepecheMood are retrieved. A score is then computed for each retrieved synset, *s*. Let *S* denotes the set of all such synsets. Two cases may appear: either the retrieved synset includes only one lemma from DepecheMood, in this case the synset gets the same score of the lemma, or, the synset includes multiple lemmas, in this case the score is the average of the scores of the corresponding lemmas. A synset, *s*, includes two sets of terms, *T*, terms that are in DepecheMood, and $\bar{T}$, terms **not in** DepecheMood.

Step 2: using the synonymy semantic relation in EWN, and based on the concept that synonym words will likely share the same emotion scores, we assign the synset scores to its corresponding terms $\bar{T}$. Again, a term *t* in $\bar{T}$ may appear in one or multiple synsets from *S*. Hence, the score assigned to *t* will be either the one of its corresponding synset or the average of the scores of its corresponding synsets that belong to *S*.

Step 3: terms in $\bar{T}$ may also appear in synsets $\bar{s}$ that do not belong to *S*. $\bar{s}$ will get the score of its corresponding terms.

Step 2 and 3 are repeated until no new terms or synsets are added and scores of added terms converged. It is important to note that we decided to consider only synonyms for expansion since synonymy is the only semantic relation that preserves the emotion orientation and does not require manual validation (Strapparava et al., 2004).

Using the described automatic expansion approach, we were able to extend the size of DepecheMood by a factor of 1.8. We obtained emotion scores for an additional 29,967 EWN terms and for 59,952 EWN synsets. Overall, we construct EmoWordNet, an emotion lexicon consisting of 67,738 EWN terms and of 59,952 EWN synsets annotated with emotion scores.

Next, we match ArSenL entries to EmoWordNet synsets. Each entry in ArSenL consists mainly of an Arabic SAMA lemma, a corresponding POS tag, a corresponding EWN synset and three sentiment scores extracted from SentiWordNet. For each entry in ArSenL, if its assigned synset is found in EmoWordNet, emotion scores of the synset are automatically added to ArSenL entry. We were able to assign emotion scores to 149,634 ArSenL entries corresponding to 32,196 Arabic lemmas, i.e., 94.71% of ArSenL lemmas. We summarize the lexicon sizes per lemma in Table 1. We also show some sample lemmas of ArSEL along their corresponding 8 emotion scores in Table 3. We have picked samples that should be emotionally charged to check if the emotions represented by the lemma have the highest scores.

As a walking example of the steps described above, we added to the steps shown in Fig. 1 an example corresponding to each step. For instance, the DepecheMood term "bonding" having noun as POS tag is mapped to EWN term "bonding" with the same POS tag. "bonding" appears in three different noun synsets in EWN with the following offset IDs: "00148653; 05665769; 13781820". Since "bonding" is the only term having a DepecheMood representation in the three synsets, the three synsets will have the same emotion scores as "bonding". While synsets "05665769; 13781820" have only the term "bonding", "00148653" includes as well the lemma "soldering" which is not in DepecheMood. Thus, from step 2, "soldering" will have the same scores as "bonding". "soldering" does not appear in any other synset so there are no more iterations. The next step is to check if the retrieved synsets appear in ArSenL. For example, "00148653" corresponds to the lemma "liHAm" and hence the Arabic lemma will be assigned the emotion scores of the synset.

To test the efficiency of our emotion lexicon ArSEL, we evaluate in the next section the performance of ArSEL when employed in emotion regression and classification tasks.

Lexicon	Lemma Count
DepecheMood	37,771
EmoWordNet	67,738
ArSenL	33,995
ArSEL	32,196

Table 1: Lexicons Coverage.

SemEval	ArSEL
Fear	Afraid
Anger	Angry
Joy	Happy
Sadness	Sad
Surprise	Inspired
Disgust	-
-	Annoyed, Amused, Don't Care

Table 2: Mapping between SemEval and ArSEL Emotion Labels.

4. ArSEL Evaluation

Since ArSEL is generated based on ArSenL, the intrinsic evaluation results of ArSenL described in (Badaro et al., 2014a) are automatically inherited by ArSEL. Therefore, we focus in this section on performing extrinsic evaluation of ArSEL. We describe next the dataset used, the experiment setup, the regression and the classification results for the two datasets: SemEval 2007 and 2018 datasets.

4.1. Using SemEval 2007 Dataset

4.1.1. About the Dataset

We utilize SemEval 2007 Affective Task dataset (Strapparava and Mihalcea, 2007). The dataset consists of one thousand news headlines annotated with six emotion scores: anger, disgust, fear, joy, sadness and surprise. For the regression task, a score between 0 and 1 is provided for each emotion. For the classification task, a threshold is applied on the emotion scores to get a binary representation of the emotions: if the score of a certain emotion is greater than 0.5, the corresponding emotion label is set to 1, otherwise it is 0. The emotion labels used in the dataset correspond to the six emotions of the Ekman model (Ekman, 1992) while those in ArSEL, EmoWordNet and DepecheMood follow the ones provided by Rappler Mood Meter. We consider the same assumptions of emotion mapping presented in the work of (Staiano and Guerini, 2014) and summarized in Table 2. Disgust emotion label in SemEval is not aligned with any emotion in EmoWordNet and hence is discarded as also assumed in (Staiano and Guerini, 2014). The dataset is in English, thus, we use Google translate to translate it automatically to Arabic. Some examples of the news' headlines along with their Google and Human translations are shown in Table 4.

4.1.2. Experiment Setup

We perform the following preprocessing steps in order to proceed with the evaluation. We utilize MADAMIRA (Pasha et al., 2014) in order to perform lemmatization for the translated dataset. The output of MADMIRA is a list of lemmas in Buckwalter transliteration (Buckwalter, 2002) along with the corresponding POS tag. We exclude lemmas that do not belong to the main four POS tags: noun, verb, adjective and adverb. It is important to note that MADAMIRA generates many fine-grained POS tags that can be grouped into the above mentioned four POS tags. On ArSEL side, we compute the average of emotion scores per lemma since an Arabic lemma can be mapped to multiple EWN synsets. Next, we compute for each news' headline the sum and the average of emotion scores. The average turned out to give better results. For the regression task, we compute Pearson correlation coefficient between the computed headline emotion scores and the scores provided in SemEval taking into consideration the mapping of emotion labels as represented in Table 2. For the classification task, we first perform min-max normalization on the computed scores and then we apply thresholding with a threshold equals to 0.5. Thus, an emotion label will be set to 1 if its corresponding emotion score is greater than 0.5, otherwise it will be set to 0. The same thresholding is applied on SemEval scores. F1 measure is then computed to evaluate classification of emotions. The experiment process is summarized in Figure 2.

4.1.3. Regression and Classification Results

We present first the coverage results of ArSEL for the translated SemEval dataset. Only one headline ("Toshiba Portege R400", "توشيبا برتجي ر٤٠٠") did not include a lemma that matched to ArSEL. In terms of lemma counts, 2,688 unique lemmas represent the dataset. 301 lemmas were not identified by MADAMIRA, 121 lemmas had POS tags different than the four main ones and 2,266 lemmas were within the four POS tags: N, V, Adj and Adv. To evaluate the coverage of ArSEL, we compare ArSEL lemmas to the 2,266 lemmas that are within the main four POS tags. 91.41% of the 2,266 lemmas were found in ArSEL. Thus, we can conclude that ArSEL includes commonly used Arabic lemmas with a high coverage.

In Table 7, Pearson correlation results are presented when using ArSEL and when using EmoWordNet on the translated SemEval Dataset and the original one respectively. We notice that the performance of ArSEL is very similar to EmoWordNet. The small difference in the scores obtained is expected since the automatic Online translation from English to Arabic cannot be guaranteed to be 100% accurate as can be seen in some of the examples shown in Table 4. Moreover, some English words may have an emotion score while their Arabic translation may not be present in ArSEL. In order to check if looking at both the English and Arabic data improves the accuracy of emotion prediction, we combine the two scores obtained from using EmoWordNet on English SemEval 2007 and

Lemma#POS	English Gloss	Afraid	Amused	Angry	Annoyed	Don't Care	Happy	Inspired	Sad
xawof#n لحام خوف	fear	**0.16866352**	0.10374394	0.13578057	0.11578797	0.09626842	0.10521568	0.12802106	0.14651883
saEAdap#n سعادة	happiness	0.01080941	0.16735222	0.01801752	0.04023918	0.18246141	**0.38946541**	0.16637971	0.02527514
taEAsap#n تعاسة	misery	0.11482094	0.11724791	0.07061617	0.13834278	0.04755821	0.1362515	0.16859612	**0.20656636**
DaHik#v ضحك	laugh	0.04837066	**0.21422647**	0.07150008	0.11078673	0.13726831	0.11134006	**0.21054358**	0.09596412
Huzon#n حزن	grief	0.01551373	0.13148076	0.0687485	0.10431947	0.06042494	0.08809219	0.21824078	**0.31317963**
$ajan#n شجن	anxiety	0.159757	0.08634377	0.10675246	0.10506455	0.11995604	0.14099477	0.05896844	**0.22216298**
maqotal#n مقتل	assault; killing	0.15997316	0.0616973	**0.33435758**	0.10675574	0.06770851	0.07205292	0.03512961	0.16232519
<izoEAj#n ازعاج	disturbance	0.05707528	0.06349826	**0.34656472**	0.14284707	0.11914421	0.11311906	0.06151737	0.096234049
kuwayĩs#a كويس	well	0.0221555	**0.24858529**	0.03319092	0.11484484	0.23663404	0.1073459	0.22167375	0.01556974
$ayo'#n شيء	thing	0.08178512	0.14615643	0.13145998	0.14008017	0.14118469	0.11244018	**0.14626546**	0.10062796

Table 3: Sample of ArSEL Arabic Lemmas with Emotion Scores.

English News' Headline	Google Translation	Human Translation
Women protest Pakistan demolition	المرأة تحتج على هدم باكستان	المرأة تحتج عل التفجير في باكستان
Dolphins, sea lions may report for duty soon	الدلافين، أسود البحر قد تقرير عن واجب قريبا	الدلافين، أسود البحر توضع في الخدمة قريباً
Woman fights to keep drunken driver in jail	امرأة تحارب للحفاظ على سائق سكران في السجن	إمرأة تحارب لإبقاء سائق سكران في السجن
Female astronaut sets record	سجل رائد فضاء أنثى	رائدة فضاء تسجل رقما قياسيا
Astronaut's arrest tests NASA's mettle	اختبارات اعتقال رائد الفضاء ناسا هزة	إعتقال رائد فضاء يضع ناسا تحت الاختبار

Table 4: News' Headlines' Examples to Show Differences between Google Translations and Human Translations.

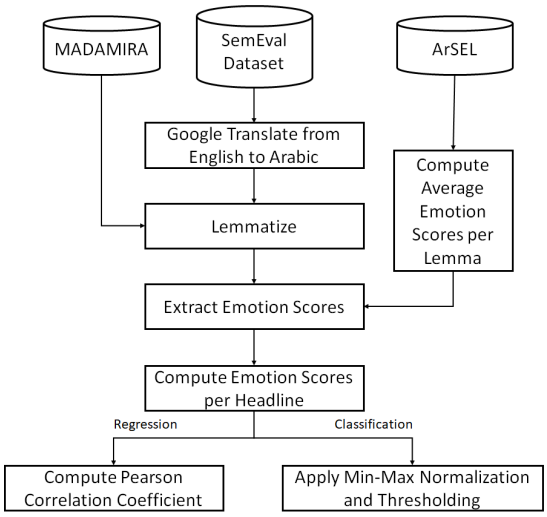

Figure 2: Overview of ArSEL Evaluation Steps.

from using ArSEL on the translated version of the same dataset. We compute the average of the two resulting scores and use it to perform regression and classification. We report the regression results in Table 7 under Combined column. As can be seen, combining the scores obtained through ArSEL and EmoWordNet improved Pearson correlation on average and consistently for all emotions except for Surprise. The discrepancy between the results achieved by EmoWordNet and ArSEL is due to the translation errors incurred by Google translate. The translation errors cause MADAMIRA to generate erroneous analysis of lemmas and hence the total emotion scores of the headline will be incorrect. The same error analysis can be inferred by looking at the other emotion classes as well.

In Table 8, we also compare F1 measure achieved by using ArSEL and EmoWordNet on translated SemEval and original one respectively. We also notice that the results for emotion classification are very close to each other. We also test the performance of combining the output of the two lexicons based on the parallel dataset shown under combined column in Table 8. Hence, we can conclude that the efficiency of EmoWordNet is preserved in ArSEL when used for emotion recognition from text. We can also deduce that emotion scores of EmoWordNet are correctly represented in ArSEL. In Table 5, we show some examples of news' headlines that were correctly classified and in Table 6, examples of news' headlines that were misclassified. By looking at the misclassified examples, we notice that misclassification is either due to predicting additional emotion labels to the actual ones (precision issue) or by predicting different emotion labels than the actual ones (recall issue). Similar to the regression task, translation errors incurred by Google translate have a negative impact on the analysis performed by MADAMIRA, thus, the translated headline is misrepresented and emotion scores assigned to the headline are incorrect.

4.2. Using SemEval 2018 Arabic Affective Tweets Dataset

While in the previous section we performed an extrinsic evaluation of ArSEL against a translated dataset from English, we present in this section an evaluation against a native Arabic dataset extracted from SemEval 2018 Task 1 "Affect in Tweets". We describe first the dataset and the coverage achieved by ArSEL and then we present results of applying regression and classification using the same approach described in section 4.1.2.

English News' Headline	Google Translation	True Emotions
Ice storms kill 21 across nation	العواصف الثلجية تقتل ٢١ عبر الأمة	fear; sadness
Thailand attacks kill three, injure 70	هجمات تايلاند قتل ثلاثة، وإصابة ٧٠	fear; sadness
Heavy snow causes travel chaos and shuts schools	تسبب الثلوج الكثيفة فوضى السفر وتغلق المدارس	fear; sadness; surprise
Israeli, Lebanese clash on border	اشتباك إسرائيلي، لبناني على الحدود	anger; fear; sadness
Catania punished for fan violence	كاتانيا يعاقب على العنف مروحة	anger; sadness

Table 5: Examples of Correctly Classified News' Headlines from SemEval 2007.

English News' Headline	Google Translation	True Emotions	Predicted Emotions
Closings and cancellations top advice on flu outbreak	إغلاق وإلغاء المشورة العليا بشأن تفشي الأنفلونزا	joy	fear; surprise
Discovered boys bring shock, joy	اكتشاف الأولاد تجلب صدمة، والفرح	joy; surprise	sadness; surprise
Iraqi sunni lands show new oil and gas promise	وتظهر الأراضي السنية العراقية وعدا جديدا للنفط والغاز	joy	fear; surprise
Golden Globes on their way	غولدن غلوب في طريقهم	joy	joy; sadness; surprise
Bush adamant on troops to Iraq	بوش يصر على القوات إلى العراق	anger; sadness	fear

Table 6: Examples of Misclassified News' Headlines from SemEval 2007.

Emotion	EmoWordNet	ArSEL	Combined
Fear	0.51	0.44	**0.53**
Anger	0.31	0.34	**0.37**
Joy	0.33	0.26	**0.35**
Sadness	**0.41**	0.31	**0.41**
Surprise	**0.17**	0.1	0.14
Average	0.35	0.29	**0.36**

Table 7: Pearson Correlation Values.

Emotion	EmoWordNet	ArSEL	Combined
Fear	0.45	**0.57**	0.55
Anger	0.17	**0.36**	**0.36**
Joy	0.48	0.55	**0.59**
Sadness	0.46	0.50	**0.55**
Surprise	0.43	0.52	**0.53**
Average	0.40	0.50	**0.52**

Table 8: F1-Measure results for emotion classification using EmoWordNet on English SemEval 2007, using ArSEL on the Arabic translated version and when combining the two scores.

4.2.1. About the Data

In SemEval 2017, a task was created for Arabic Twitter sentiment analysis (Rosenthal et al., 2017). Several teams participated and the winning teams were NileTMRG (El-Beltagy et al., 2017) and OMAM (Baly et al., 2017b; Onyibe and Habash, 2017). In SemEval 2018, the focus was on Emotion classification from text. We utilize the provided competition dataset to evaluate ArSEL. SemEval 2018 dataset consists of Arabic tweets that are annotated with four emotions: anger, fear, joy and sadness along with the intensity present for each one. We have only access to the training and the development sets. In total, there are 2,871 tweets. In Table 9, we show the distribution of emotions across the tweets. The frequencies of the emotions are very close to each other with "Sadness" being the most frequent in the dataset. We follow the same experiment setup described in section 4.1.2, but we do not need the translation part since the data is already in Arabic. Instead, we perform additional preprocessing steps given that the dataset is extracted from Twitter. We clean the tweets from the hash tag and the underscore characters. We then feed the tweets to MADAMIRA to extract lemmas. In terms of ArSEL coverage, we were able to match 83.47% of the generated lemmas that belong to one of the four main POS tags. We were not able to generate any emotion scores for three tweets that mainly consisted of dialectal Arabic terms (عيونج , your eyes) elongations (خاااااف, fear) and emoticons.

Emotion	Number of Occurrence
Fear	1028
Anger	1027
Joy	952
Sadness	**1030**

Table 9: Distribution of Emotion Labels across the Tweets.

4.2.2. Regression and Classification Results

We follow the same approach described in section 4.1.2 to perform regression and classification with the modifications described in section 4.2.1. We use the average of the scores of the four emotions (joy, fear, anger and sadness), mutually present in ArSEL and in SemEval 2018 dataset. We have tried the sum of the emotions' scores as well, but, using average showed to be better. For the regression, we evaluate Pearson correlation coefficient against the intensity scores provided in the Twitter data. On average, we achieve an R score of 0.26. Table 12 shows the results per emotion.

For classification, we also apply min-max normalization and compare against the provided labels in the data. We use F1 measure as an evaluation metric. We also compare the results of our naïve unsupervised classifier to a majority baseline classifier where the predictor will always assign "Sadness" to the tweet since it is the most frequent emotion. The results are shown in Table 13. We outperform the baseline by an average of 30% as F1-score. Thus, we can confirm the efficiency of using ArSEL for emotion recognition tasks. We expect better results to be

Arabic Tweet	Translation	True Emotions
عيد سعيد جدا ما نعرف كلام كفار	A very happy holiday we don't know words of unbelievers	joy
!!اليش هالفتره عباره عن دموع وحرقت قلب؟	Why is this period full of tears and sorrow	sadness
يارب هالفجر جايب معاه كل خير وسعاده وتوفيق	Oh God, this dawn is bringing all good, happiness and reconciliation	joy
ماعليك زود يالجوهرة كلك لطف وحبابة يسعدك ربي	You are a diamond of niceness and loveliness, may God make happy	joy
صباح الخير و الفرح بيوم المرأة العالمي انشالله أيامك كلها أعياد مايا	Good morning and joy in international Women's day	anger; joy

Table 10: Examples of Correctly Classified Arabic Tweets from SemEval 2018.

Arabic Tweet and English Translation	True Emotions	Predicted Emotions
كنت اكتر انسانه بتخاف بس صارلي كم سنه كتير جريئه حتى بحضر افلام رعب لحالي و عادي I used to be scared from watching scary movies but I have been watching them by myself since a while	joy	fear
هرفي لو اعطوني ببلاش ما اخذت من عندهم شي ، مرتين دخلت المستشفى بسببهم Even if they gave it for free I won't take it, I was admitted to the hospital twice because of them	fear	anger; sadness
عشان كده العرب كانو بيشوفو ان اللي عيونهم ملونه نذير شؤم تحسي عينه فيها غدر وشر That's why Arabs thought that people with colored eyes are evil	fear	joy; sadness
شيء ما يقوم بإشعال فتيل الرهبة في قلبي كلما تعلق الموضوع بالحب I have fear feelings whenever the subject is related to love	fear	sadness
طيب طالما هوا عتاب كيف صار فراق ؟؟ Since it was reproach why did it become separation?	sadness	joy

Table 11: Examples of Misclassified Arabic Tweets from SemEval 2018.

Emotion	R Value
Fear	0.26
Anger	0.25
Joy	0.31
Sadness	0.22
Average	0.26

Table 12: Pearson Correlation Results on SemEval 2018 Arabic Tweets Dataset.

Emotion	ArSEL	Majority Baseline
Fear	0.32	0
Anger	0.41	0
Joy	0.52	0
Sadness	0.46	0.5
Average	**0.43**	0.13

Table 13: Classification F1-score Results on SemEval 2018 Arabic Tweets Dataset.

achieved when utilizing more sophisticated regression and classification techniques.

We also show examples of correctly classified tweets in Table 10, whereas in Table 11, we present examples of misclassified tweets.

By analyzing some of the misclassification examples we can see that several tweets are in dialectal Arabic which may produce erroneous morphological analysis. Moreover, some words have different meanings and emotion significance especially when used in dialectal Arabic such as the word "طيب" which could mean good, ok, tasty or alright. Last but not least, it is important to have a comprehensive model that takes into consideration the whole tweet rather than only word components as for instance in the first example in Table 11: although the words "خاف" and "رعب", which relate to fear are present in the tweet, the overall emotion is joy since the writer is happy that she has overcome her fear and she has been able to watch scary movies without any problem.

5. Conclusion and Future Work

We presented in this paper ArSEL, a large scale Arabic Sentiment and Emotion Lexicon. ArSEL is constructed automatically by using three lexical resources: DepecheMood, English WordNet and ArSenL. First, DepecheMood is mapped to EWN. Then, it is expanded iteratively using EWN synonymy semantic relation. The resulting expanded version of DepecheMood, EmoWordNet, is then linked to ArSenL entries using EWN synset IDs that exist in both lexicons. ArSEL consists of 32,196 Arabic lemmas annotated simultaneously with sentiment and emotion scores. ArSEL will be made publicly available on `http://oma-project.com` to speed up research in the area of emotion recognition from text. Moreover, using ArSEL in emotion classification task proved to be efficient with comparable performance to when utilizing EmoWordNet on an English dataset. Using ArSEL in a simplistic classification model outperformed a majority baseline predictor by 30% in terms of F1 measure. As future work, we would like to investigate more complex and sophisticated emotion recognition models and test the proposed models on larger datasets.

6. Bibliographical References

Abdaoui, A., Azé, J., Bringay, S., and Poncelet, P. (2017). Feel: a french expanded emotion lexicon. *Language Resources and Evaluation*, 51(3):833–855.

Abdul-Mageed, M. and Diab, M. (2012). Toward building a large-scale Arabic sentiment lexicon. In *Proceedings of the 6th International Global WordNet Conference*, pages 18–22.

Abdul-Mageed, M. and Diab, M. T. (2014). Sana: A large scale multi-genre, multi-dialect lexicon for Arabic subjectivity and sentiment analysis. In *LREC*, pages 1162–1169.

Abdul-Mageed, M. and Ungar, L. (2017). Emonet: Fine–grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 718–728.

Abdul-Mageed, M. (2017). Modeling Arabic subjectivity and sentiment in lexical space. *Information Processing & Management*.

Al Sallab, A. A., Baly, R., Badaro, G., Hajj, H., El Hajj, W., and Shaban, K. B. (2015). Deep learning models for sentiment analysis in arabic. In *ANLP Workshop*, volume 9.

Al-Sallab, A., Baly, R., Hajj, H., Shaban, K. B., El-Hajj, W., and Badaro, G. (2017). Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):25.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Badaro, G., Hajj, H., El-Hajj, W., and Nachman, L. (2013). A hybrid approach with collaborative filtering for recommender systems. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 349–354. IEEE.

Badaro, G., Baly, R., Hajj, H., Habash, N., and El-Hajj, W. (2014a). A large scale Arabic sentiment lexicon for Arabic opinion mining. *ANLP 2014*, 165.

Badaro, G., Baly, R., Hajj, H., Habash, N., El-hajj, W., and Shaban, K. (2014b). An efficient model for sentiment classification of Arabic tweets on mobiles. In *Qatar Foundation Annual Research Conference*, number 1, page ITPP0631.

Badaro, G., Hajj, H., Haddad, A., El-Hajj, W., and Shaban, K. B. (2014c). A multiresolution approach to recommender systems. In *Proceedings of the 8th Workshop on Social Network Mining and Analysis*, page 9. ACM.

Badaro, G., Hajj, H., Haddad, A., El-Hajj, W., and Shaban, K. B. (2014d). Recommender systems using harmonic analysis. In *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*, pages 1004–1011. IEEE.

Badaro, G., Baly, R., Akel, R., Fayad, L., Khairallah, J., Hajj, H., Shaban, K., and El-Hajj, W. (2015). A light lexicon-based mobile application for sentiment mining of arabic tweets. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 18–25.

Baly, R., Badaro, G., El-Khoury, G., Moukalled, R., Aoun, R., Hajj, H., El-Hajj, W., Habash, N., and Shaban, K. (2017a). A characterization study of arabic twitter data with a benchmarking for state-of-the-art opinion mining models. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 110–118.

Baly, R., Hajj, H., Habash, N., Shaban, K. B., and El-Hajj, W. (2017b). A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in arabic. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):23.

Bandhakavi, A., Wiratunga, N., Deepak, P., and Massie, S. (2014). Generating a word-emotion lexicon from# emotional tweets. In ** SEM@ COLING*, pages 12–21.

Bandhakavi, A., Wiratunga, N., Massie, S., and Padmanabhan, D. (2017). Lexicon generation for emotion detection from text. *IEEE intelligent systems*, 32(1):102–108.

Bernard, J. R. L.-B. and Bernard, J. R. L.-B. (1986). *The Macquarie Thesaurus*. Macquarie.

Black, W., Elkateb, S., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A., and Fellbaum, C. (2006). Introducing the Arabic wordnet project. In *Proceedings of the third international WordNet conference*, pages 295–300.

Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of computational science*, 2(1):1–8.

Bougie, R., Pieters, R., and Zeelenberg, M. (2003). Angry customers don't come back, they get back: The experience and behavioral implications of anger and dissatisfaction in services. *Journal of the Academy of Marketing Science*, 31(4):377–393.

Brave, S. and Nass, C. (2003). Emotion in human–computer interaction. *Human-Computer Interaction*, page 53.

Buckwalter, T. (2002). Buckwalter {Arabic} morphological analyzer version 1.0.

Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., Lee, S., Neumann, U., and Narayanan, S. (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 205–211. ACM.

Cambria, E., Havasi, C., and Hussain, A. (2012). Senticnet 2: A semantic and affective resource for opinion mining and sentiment analysis. In *FLAIRS conference*, pages 202–207.

Constantine, L., Badaro, G., Hajj, H., El-Hajj, W., Nachman, L., BenSaleh, M., and Obeid, A. (2016). A framework for emotion recognition from human computer interaction in natural setting. *22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016), Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM 2016)*.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J. G. (2001). Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, 18(1):32–80.

Das, D., Poria, S., and Bandyopadhyay, S. (2012). A classifier based approach to emotion lexicon construction. In *NLDB*, pages 320–326. Springer.

Ekman, P. (1992). An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

El-Beltagy, S. R., Kalamawy, M. E., and Soliman, A. B. (2017). Niletmrg at semeval-2017 task 4: Arabic sentiment analysis. *arXiv preprint arXiv:1710.08458*.

El Gohary, A. F., Sultan, T. I., Hana, M. A., and El Dosoky, M. (2013). A computational approach for analyzing and detecting emotions in Arabic text. *International Journal of Engineering Research and Applications (IJERA)*, 3:100–107.

Esuli, A. and Sebastiani, F. (2007). Sentiwordnet: A high-coverage lexical resource for opinion mining. *Evaluation*, pages 1–26.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any–domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Fellbaum, C. (2010). Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

Fragopanagos, N. and Taylor, J. G. (2005). Emotion recognition in human–computer interaction. *Neural Networks*, 18(4):389–405.

Goldman, A. I. and Sripada, C. S. (2005). Simulationist models of face-based emotion recognition. *Cognition*, 94(3):193–213.

Graff, D., Maamouri, M., Bouziri, B., Krouna, S., Kulick, S., and Buckwalter, T. (2009). Standard Arabic morphological analyzer (sama) version 3.1. *Linguistic Data Consortium LDC2009E73*.

Gunes, H. and Piccardi, M. (2007). Bi-modal emotion recognition from expressive face and body gestures. *Journal of Network and Computer Applications*, 30(4):1334–1345.

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

Hibbeln, M., Jenkins, J. L., Schneider, C., Valacich, J. S., and Weinmann, M. (2017). How is your user feeling? inferring emotion through human–computer interaction devices. *MIS Quarterly*, 41(1).

Houjeij, A., Hamieh, L., Mehdi, N., and Hajj, H. (2012). A novel approach for emotion classification based on fusion of text and speech. In *Telecommunications (ICT), 2012 19th International Conference on*, pages 1–6. IEEE.

Jaimes, A. and Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1):116–134.

Knautz, K., Siebenlist, T., and Stock, W. G. (2010). Memose: search engine for emotions in multimedia documents. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and development in information retrieval*, pages 791–792. ACM.

Liu, H. and Singh, P. (2004). Conceptnet: a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Liu, B. and Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer.

Mohammad, S. M. and Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Mohammad, S. M. and Yang, T. W. (2011). Tracking sentiment in mail: How genders differ on emotional axes. In *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis*, pages 70–79. Association for Computational Linguistics.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Mohammad, S. M. (2017). Word affect intensities. *arXiv preprint arXiv:1704.08798*.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2007). Textual affect sensing for sociable and expressive online communication. *Affective Computing and Intelligent Interaction*, pages 218–229.

Neviarouskaya, A., Prendinger, H., and Ishizuka, M. (2011). Sentiful: A lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, 2(1):22–36.

Onyibe, C. and Habash, N. (2017). Omam at semeval-2017 task 4: English sentiment analysis with conditional random fields. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 670–674.

Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

Pantic, M. and Rothkrantz, L. J. (2003). Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE*, 91(9):1370–1390.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *LREC*, volume 14, pages 1094–1101.

Patwardhan, A. S. and Knapp, G. M. (2017). Multimodal affect analysis for product feedback assessment. *arXiv preprint arXiv:1705.02694*.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. *Theories of emotion*, 1(3-31):4.

Plutchik, R. (1994). *The psychology and biology of emotion.* HarperCollins College Publishers.

Poria, S., Gelbukh, A., Das, D., and Bandyopadhyay, S. (2012). Fuzzy clustering for semi-supervised learning–case study: Construction of an emotion lexicon. In *Mexican International Conference on Artificial Intelligence*, pages 73–86. Springer.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Shaheen, S., El-Hajj, W., Hajj, H., and Elbassuoni, S. (2014). Emotion recognition from text based on automatically generated rules. In *Data Mining Workshop (ICDMW), 2014 IEEE International Conference on*, pages 383–392. IEEE.

Staiano, J. and Guerini, M. (2014). Depechemood: A lexicon for emotion analysis from crowd-annotated news. *arXiv preprint arXiv:1405.1605*.

Stone, P. J., Dunphy, D. C., and Smith, M. S. (1966). The general inquirer: A computer approach to content analysis.

Strapparava, C. and Mihalcea, R. (2007). Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics.

Strapparava, C., Valitutti, A., et al. (2004). Wordnet affect: an affective extension of wordnet. In *LREC*, volume 4, pages 1083–1086.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Trad, C., Hajj, H. M., El-Hajj, W., and Al-Jamil, F. (2012). Facial action unit and emotion recognition with head pose variations. In *ADMA*, pages 383–394. Springer.

Wang, W., Chen, L., Thirunarayan, K., and Sheth, A. P. (2012). Harnessing twitter" big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 587–592. IEEE.

Wegrzyn, M., Vogt, M., Kireclioglu, B., Schneider, J., and Kissler, J. (2017). Mapping the emotional face. how individual face parts contribute to successful emotion recognition. *PloS one*, 12(5):e0177239.

Xu, G., Meng, X., and Wang, H. (2010). Build chinese emotion lexicons using a graph-based algorithm and multiple resources. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1209–1217. Association for Computational Linguistics.

Yang, C., Lin, K. H.-Y., and Chen, H.-H. (2007). Building emotion lexicon from weblog corpora. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 133–136. Association for Computational Linguistics.

ArSentD-LEV: A Multi-Topic Corpus for Target-based Sentiment Analysis in Arabic Levantine Tweets

Ramy Baly(1), **Alaa Khaddaj**(2), **Hazem Hajj**(2), **Wassim El-Hajj**(3), **Khaled Bashir Shaban**(4)
(1) MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
(2) American University of Beirut, Electrical and Computer Engineering Department, Beirut, Lebanon
(3) American University of Beirut, Computer Science Department, Beirut, Lebanon
(4) Qatar University, Computer Science and Engineering Department, Doha, Qatar
baly@mit.edu, {awk11, hh63}@aub.edu.lb, we07@aub.edu.lb, khaled.shaban@qu.edu.qa

Abstract

Sentiment analysis is a highly subjective and challenging task. Its complexity further increases when applied to the Arabic language, mainly because of the large variety of dialects that are unstandardized and widely used in the Web, especially in social media. While many datasets have been released to train sentiment classifiers in Arabic, most of these datasets contain shallow annotation, only marking the sentiment of the text unit, as a word, a sentence or a document. In this paper, we present the Arabic Sentiment Twitter Dataset for the Levantine dialect (ArSenTD-LEV). Based on findings from analyzing tweets from the Levant region, we created a dataset of 4,000 tweets with the following annotations: the overall sentiment of the tweet, the target to which the sentiment was expressed, how the sentiment was expressed, and the topic of the tweet. Results confirm the importance of these annotations at improving the performance of a baseline sentiment classifier. They also confirm the gap of training in a certain domain, and testing in another domain.

Keywords: Corpus development, Levantine tweets, multi-topic, sentiment analysis, sentiment target

1. Introduction

Sentiment analysis refers to the task of inferring opinions from text (Liu, 2012). Research in sentiment analysis has been driven by the interest in its wide range of applications and the availability of large amounts of subjective data on the Web (Ravi and Ravi, 2015). Today's social media has provided people the opportunity to connect across the globe and express their opinions and emotions freely and abundantly. Twitter is one of the most used social media platforms, with recent statistics [1] indicating that over 500 million tweets are being sent out daily, mainly to express opinions about personal or trending topics, news or events (Sareah, 2015).

Sentiment analysis has been widely approached as a text classification problem, with the target of predicting the overall opinion of a given text (words, sentences or documents) (Pang et al., 2002; Socher et al., 2013; Tang et al., 2015; Farra et al., 2010). However, sentiment analysis can also be performed at more granular levels, such as identifying target entities (Brody and Elhadad, 2010; Somasundaran and Wiebe, 2009; Farra and McKeown, 2017) and predicting opinions towards these targets, whether in Twitter (Jiang et al., 2011), online comments (Biyani et al., 2015) or product reviews (Wang et al., 2016; Kirange and Deshmukh, 2015). These tasks are critically-important to handle cases where the text contains multiple opinions expressed towards one or different targets, which is a common phenomenon in product reviews.

Research in exploring methods for English sentiment analysis has been leading the way, while other languages, including Arabic, still lag behind. Most advances were made in English, mainly because to the availability of sentiment corpora to support such tasks. This paper aims at providing new resources to support research advances in Arabic. As a matter of fact, Arabic ranks as the 4th most spoken language worldwide (Paolillo and Das, 2006), and as of March 2017, 11.1 million Twitter users from the Arab world are generating 27.4 million tweets on a daily basis (Salem, 2017).

In the last few years, there has been a significant progress in creating resources for Arabic sentiment analysis. However, these resources are often coupled with sentiment annotations only, and typically on a three point scale (1 to 3) instead of the common 5-point typically used in reviews, which also reflects sentiment intensity. Furthermore, it was found that modeling sentiment depends on the domain or topic at hand, and that a sentiment model trained on one domain is not expected to perform as well on another (Pan et al., 2010). Additionally, textual semantics vary across languages and dialects (Baly et al., 2017a) due to cultural factors (Salameh et al., 2015; Mohammad et al., 2016). For example, سبحَان اللّه العلي العظيم *Glory to God the Great* is used in the Levant to express positive sentiment, whereas it is considered a religious saying with no sentiment in other Arab regions, *e.g.* the Gulf countries. Consequently, cross-lingual and cross-domain approaches (Chen et al., 2016; Li, 2017) have been explored to avoid the need for a sentiment corpus for each domain or language, which is costly and time-consuming.

In this paper, we address the limitations of having a corpus annotated for sentiment only, by creating a corpus and having it simultaneously annotated for different and important aspects needed for research in sentiment analysis. We create our corpus from Twitter content due to its widespread use in the Arab world. Given the cultural and linguistic differences across Arab regions, causing shifts in semantics, we focus on developing sentiment models for the Levantine dialect. According to (Zaidan and Callison-Burch, 2014), Arabic dialects can be categorized into Egyptian, Levantine, Gulf, Iraqi and Maghrebi. Our corpus is composed of tweets retrieved from Levantine countries (Jordan, Lebanon, Palestine and Syria), where the Levantine dialect is the 3rd most spoken Arabic dialect (Zaidan and Callison-

[1] https://www.socialbakers.com/statistics/twitter/

Burch, 2014). We selected a group of 4,000 tweets, and had users annotate those tweets via crowdsourcing to: 1) identify the sentiment targets in each tweet, 2) annotate both sentiment polarity and intensity on a five-point scale, from *very negative* to *very positive*, 3) indicate whether sentiment was expressed *implicitly* or *explicitly*, and 4) finally to identify the topic the tweet is discussing. This corpus is publicly available. [2]

The resulting corpus provides a resource complement to existing Arabic dialect resources (Baly et al., 2017c; Assiri et al., 2016; Refaee and Rieser, 2014a). It will also enable models that can exploit sentiment target identification, topic identification and sentiment expression. Furthermore, it will opens doors to investigate cross-dialect sentiment models by leveraging existing Twitter corpora from other regions and dialects. Several experiments are conducted to confirm the benefits of such new aspects (Joty et al., 2017). We show that topic-based models outperform models that do not consider the topic of the text.

The remaining of the paper is organized as follows. Section 2 describes previous efforts to create sentiment datasets in Arabic. Section 3 presents an analysis of Arabic tweets and describes our methodology to create and annotate the corpus. Section 4 presents experimental results to benchmark the performance of a baseline classifier on our developed corpus, and also to emphasize the impact of topic change on the performance. Concluding remarks are made in Section 5.

2. Related Work

Sentiment analysis has been performed by training machine learning models using different choices of features (Abdul-Mageed et al., 2011; Abdul-Mageed et al., 2014; Badaro et al., 2014; Refaee and Rieser, 2014b; Badaro et al., 2015; Al Sallab et al., 2015; Baly et al., 2016; Baly et al., 2017b; Al-Sallab et al., 2017). However, training and evaluating accurate sentiment models requires the availability of corpora with sentiment labeling. Below, we list commonly-known Arabic sentiment corpora.

Abdul-Mageed et al. (2011) created a corpus by annotating 2,855 sentences, coming from the first 400 documents of the Penn Arabic Treebank Version 1 Part 3 (Maamouri et al., 2004), using the following labels: objective, subjective-positive, subjective-negative and subjective-neutral. This dataset was extended by annotating additional 5,342 sentences from Wikipedia talk pages and 2,532 sentences from web forums to create the AWATIF corpus (Abdul-Mageed and Diab, 2012). Rushdi-Saleh et al. (2011) created the Opinion Corpus for Arabic (OCA), which consists of 500 Arabic movie reviews that are annotated as either positive or negative. Aly and Atiya (2013) created LABR; a large-scale corpus consisting of 63,257 book reviews written in Arabic, each rated on a five-point scale. ElSahar and El-Beltagy (2015) retrieved 33,116 Arabic reviews on movies, hotels, restaurants and products, and automatically annotated them using available ratings.

The above-mentioned corpora contained data written in Modern Standard Arabic (MSA). Additional efforts have been made to develop corpora for dialectal Arabic, due to its widespread use in the Web. Refaee and Rieser (2014a) retrieved 8,868 tweets from multiple Arabic dialects, and annotated them for both subjectivity and sentiment using the following labels: polar, positive, negative, neutral and mixed. Baly et al. (2017d) created the Arabic Sentiment TreeBank (ArSenTB) using 1,176 comments, from the QALB dataset (Mohit et al., 2014), written in MSA and a mixture of different dialects. In addition to sentence-level sentiment annotation, comments were transformed into phrase structure parse trees, and the sentiment of each constituent (node in the tree) was also annotated, totaling up to 123,000 constituents. Al-Kabi et al. (2016) created a corpus covering MSA as well as several Arabic dialects. This corpus is composed of 1,442 reviews extracted from five domains: economy, food-life style, religion, sports and technology. Annotation was performed manually to ensure high quality. Nabil et al. (2015) created the Arabic Sentiment Tweets Dataset (ASTD), which consists of 10,006 tweets, written in the Egyptian dialect and annotated as positive (799), negative (1,684), mixed (832) or objective (6,691). Medhaffar et al. (2017) created the Tunisian Sentiment Analysis Corpus (TSAC) by retrieving 17,000 comments written with Tunisian dialect from Facebook, and annotating them as positive or negative. Baly et al. (2017a) created two datasets, each consisting of 1,000 tweets, written in Egyptian and Emarati dialects and manually annotated for sentiment at a 5-point scale, from very negative to very positive. A similar effort was done to create AraSenti-Tweet; a sentiment corpus of 17,573 tweets written in MSA and in Saudi dialect (Al-Twairesh et al., 2017).

It can be observed that, despite the recent efforts to create Arabic sentiment corpora, the majority of these datasets only focused on labeling the overall sentiment of the text, while ignoring other useful information, such as the target of the sentiment and the topic being discussed. A corpus with similar annotations was developed for SemEval-2016 Task 4 on Sentiment Analysis in Arabic tweets (Rosenthal et al., 2017). The corpus consisted of 3,355 tweets annotated by the polarity of sentiment in the tweet and the sentiment towards a specific target in the tweet (also known as stance). Also, (Al-Smadi et al., 2015) used a subset of 2,800 reviews from the LABR corpus and enriched it with aspect-based sentiment annotations.

In this paper, we present ARSENTD-LEV; an Arabic sentiment dataset that is composed of Levantine tweets, and we enrich it with a variety of sentiment-related annotations that never existed together in a single corpus.

3. Dataset

In this section, we describe our methodology to create the new sentiment corpus.

3.1. Manual Data Analysis

To have the proper guidelines in the annotation process, we conducted manual analysis to make sure we have solid insights into the intricacies of the sentiment analysis and the required sentiment annotations. The goal of the analysis was to gain insights and understand the characteristics and different usages of Twitter in the Levant region. As such, a

[2]The corpus is available at `www.oma-project.com`

sample of 200 tweets, generated in countries from the Levant region, were retrieved and characterized. We focused on information that should be critical to developing accurate sentiment analysis models, including: the topic being discussed, the language being used, the way sentiment was being expressed and the target of the sentiment.

Topic Analysis The first question we wanted to answer is: *what topics are often discussed on Twitter?*. Our findings, shown in Table 1, suggest that most of the tweets expressed opinions about personal and daily matters, and to a less extent on political issues, especially the ongoing conflicts in the Middle East. People also discussed religious matters and tend to quote verses from the Quran. Table 1 also illustrates the different items discussed per topic, ordered from most to least frequent in the sample set. In addition to the outcome of knowing which topics were being discussed, we also used the sample tweets to identify the most discriminative keywords across topics, which are used later when creating the corpus.

Topic	Size	Sub-topics
Personal	36%	sarcasm, love, sadness and optimism
Politics	23%	Syrian war, Palestinian war, Lebanese elections, revolution and terrorism
Religion	11%	sermon, mention, praising God, religious events and Quranic verses
Sports	6%	international and local soccer games, soccer players and basketball
Other	24%	entertainment, ads, health, education, economy, technology and weather

Table 1: Breakdown of the different topics and sub-topics that were discussed in the sample set of 200 tweets.

Language Use By analyzing the language that was used to write the 200 tweets, we observed that: 51% were written in Modern Standard Arabic (MSA), 34% in Levantine dialect, and the remaining 15% in English, Arabizi, or a mixture of MSA and dialectal Arabic (DA). We also observed that most personal tweets were written in DA, indicating that users prefer to use it rather than MSA when it comes to discussing personal aspects of their lives and feelings.

Sentiment Expression We analyzed the sentiment distribution in the 200 tweets by labeling the sentiment polarity and the way it was expressed, i.e., explicitly or implicitly, for each tweet. We observed that a significant amount of tweets were negative, which can be attributed to the current political situation having a direct impact on people's lives and opinions. We also observed that sentiment distribution changes from one country to another; it is mostly negative in Syria and mostly neutral in Jordan, which may reflects the countries' political and social stabilities. Finally, among the subjective tweets, sentiment was expressed explicitly in 64% and implicitly in 35% of the tweets, which is an indication of the complexity in opinion mining.

3.2. Corpus Development

Our goal is to create an Arabic dataset of tweets from the Levant region, and annotate them for topic, sentiment polarity, sentiment intensity, sentiment target and sentiment expression. In order to create this corpus we performed the following steps.

Tweets Retrieval We used the TWEEPY python module to retrieve tweets using pre-specified geo-locations covering four countries from the Levant region: Jordan, Lebanon, Palestine and Syria. The retrieval process began on November 5th 2017 and ended on November 29th 2017. As a result, we retrieved 45,000 tweets that are equally distributed across the four countries.

Pre-processing The target size of our corpus is 4,000 tweets; 1,000 for each country. We also aim to collect tweets discussing the common topics (politics, religion, sports, personal and entertainment) that we encountered in the manual analysis. Therefore, we created for each of the five topics a list of topic-specific keywords; for each topic we selected the most frequent words in the sample set that were the most discriminative with regard to that topic. We checked the 45K tweets against these lists and kept those that contained at least one keyword from one list and none from the others. This is a naive topic classification that will not be part of the final corpus, and that was performed only to increase the likelihood of having tweets discussing our target topics. We also excluded tweets written in foreign languages and those only containing URLs and emoticons. Finally, for each country, we selected the longest 1,000 tweets such that they are balanced across our target topics. It is worth mentioning that despite the fact that we enforced some balance over the different topics, we do not expect this to be the case in the final corpus after manual annotation, since topics are inherently imbalanced as shown in Table 1.

Annotation The annotation process was carried out via crowdsourcing and using the CrowdFlower platform. For each tweet, annotators were instructed to 1) select its overall sentiment, 2) identify the target of this sentiment in the tweet (in case it was not neutral) by copying segments of the tweet into a text box, 3) identify whether the sentiment was expressed explicitly or implicitly, and 4) specify the topic being discussed. Sentiment labels were assigned based on a 5-point scale using the following labels: *very negative*, *negative*, *neutral*, *positive* and *very positive*. Motivated by our manual analysis of a sample of tweets, we pre-defined the following topics: *politics*, *religions*, *sports* and *personal*. If a tweet's topic did not belong to one of these choices, annotators will have to specify another topic based on their own judgment. Before conducting the large-scale annotation task, we conducted a pilot task to ensure the clarity of the guidelines and examples, and consequently the task.
Tweets were randomly assigned to at least 5 annotators, and up to 4 additional annotators were asked to participate in case of ties. As a result, we had 5-9 different annotators an-

Topic		Sentiment		Expression	
Personal	32.6%	Very negative	16.3%	Explicit	73.6%
Sports	12.12%	Negative	30.8%	Implicit	4.3%
Politics	37.63%	Neutral	22.13%	None	22.1%
Religions	9.83%	Positive	20.1%		
Entertainment	4.35%	Very positive	10.7%		
Other	3.45%				

Table 2: Distributions of the different annotated features in the corpus.

notating each tweet, which is a reasonable number to perform aggregation over 5 classes. Annotations were aggregated based on majority voting, and the annotators' trust score (reflecting their work accuracy) was used for breaking ties. To make sure only qualified annotators are allowed to do the task, we performed quality control by creating a gold set of 181 tweets that we annotated for sentiment, and used it to monitor the annotators' accuracy on this set. Only those with an accuracy higher than 75% were allowed to stay on the job.

Post-Processing To aggregate sentiment targets returned by annotators, we automatically extracted the longest common substring among targets whose annotators agreed with the final aggregated label. In other words, if the aggregated sentiment was *positive*, we only considered the pool of targets returned by annotators who annotated the tweet as either *positive* or *very positive*. Also, while we instructed annotators that the sentiment target must be explicitly observed in the tweet, we observed that in 160 tweets, annotators specified the targets with their own wording. We resolved these cases manually. We also manually aggregated the topic annotations of 138 tweets whose topic was not one of the pre-specified topics.

3.3. Statistics and Evaluation

It is of critical importance to evaluate the annotation quality to make sure the corpus can be properly used to develop accurate sentiment models. We evaluated how well annotators of the each tweet agreed on the same label. Over a sample of 100 tweets, the average agreement was 83% for topics, 73% for sentiments, and 72% for sentiment expressions. These numbers are significantly higher than 50% (the case of a tie), indicating a straightforward majority-based aggregation for most of the tweets. Differences in agreements reflect the relative difficulty of the task. For instance, it can be inferred that identifying the sentiment of a tweet and how it was expressed is a more ambiguous and subjective task than identifying the topic. It is worth mentioning that the agreement on sentiment increases up to 81% when considering three sentiment classes, which indicates that many cases of disagreement were due to differences in annotating the intensity.

We also report a 83% agreement between the labels of the gold set (181) tweets, and the aggregation of the CrowdFlower-annotated sentiments for the same tweets. In order to evaluate the quality of sentiment targets, we manually annotated the targets for the gold set of tweets, and compared them to the outcome of selecting the longest common substring among CrowdFlower-annotated targets for the same tweets. By counting the number of common words between both targets and normalizing it by the length of the gold target, we found a 63% overlap, on average, which is acceptable given the highly-subjective nature of the task. Finally, statistics and distributions of the different annotated features from the corpus are presented in Table 2.

4. Experiments and Results

In this section, we present the results of applying a baseline sentiment classifier on our new corpus: ArSenTD-LEV. We also perform cross-topic and in-topic experiments to emphasize the impact of changing the topic between training and testing data, and also by using the *topic* and *sentiment expression* as additional features to train the classifier.

Our feature set is composed of *uni*-grams and *bi*-grams represented with TF-iDF scores. These features were used to train different classifiers including logistic regression, Support Vector Machines (SVM), random forest trees and the ridge classifier. We report only the results of logistic regression, which achieved better results. Results are reported using accuracy and F1 score averaged across the different classes (Macro-F1).

First, we train a generic model on the whole corpus with 5-fold cross-validation. In this case, the model is trained on different topics and dialects. We show in Table 3 that, by only adding the *country*, *topic* and *sentiment expression* features directly from the corpus, the performance significantly increases by 13 absolute points. This indicates the importance of these features for sentiment analysis, and relates back to our manual analysis in which we found sentiment variations across topics and dialects.

We also highlight the impact of change-of-topic between training and testing by conducting two experiments. In the first experiment, we train our model and test it on data from the same topic, i.e., the *topic* feature is implicitly embedded in the model. In the second experiment, we train our model on data from one topic and test on data from another topic. We also evaluate, for each experiment, the impact of adding the *sentiment expression* feature. We perform these experiments on the *politics* and *personal* domains, which are the most frequent topics in our corpus. We create fixed sets for training and testing with equal sizes in both topics, and use the same splits for all experiments.

Results in Table 3 show a significant drop in accuracy due to the change-of-topic from training to testing. This is a typical problem seen when developing cross-domain sentiment models instead of training topic-specific models,

Features		Generic	Same-Topic		Cross-Topic	
		cross-val	*Politics*	*Personal*	*Pol-Pers*	*Pers-Pol*
uni/bi-grams	Acc.	0.51	0.58	0.40	0.31	0.36
	Macro-F1	0.50	0.53	0.39	0.21	0.29
uni/bi-grams + annotations	Acc.	0.63	0.64	0.56	0.47	0.50
	Macro-F1	0.63	0.62	0.54	0.37	0.44

Table 3: Experimental results of a baseline logistic regression model showing the impact of adding the corpus annotation features, and the impact of changing the topic from training to testing.

which is an expensive solution. Our corpus allows the development of models for domain adaptation given the availability of topic annotation. Results also confirm the importance of the *sentiment expression* feature, which alone helped improving the performance by more than 10% absolute. It can be observed that results on the *personal* domain are much lower than those in the *politics* domain, which can be attributed to the wider range of sub-topics that can be covered by the *personal* domain.

5. Conclusion

In this paper, we presented the ArSenTD-LEV; a corpus for sentiment analysis in Arabic Levantine tweets. Based on a manual analysis that we conducted on a sample of 200 tweets retrieved from the Levant region, we realized the importance of knowing: 1) the topic being discussed by the tweet, 2) the target to which the sentiment was expressed, and 3) the manner the sentiment was expressed, to predict the sentiment of the tweet more accurately. Consequently, our developed corpus consists of 4,000 tweets collected from Levantine countries (Jordan, Lebanon, Palestine and Syria). For each tweet, the corpus specifies its overall sentiment, the target to which that sentiment was expressed, and how it was expressed (explicitly or implicitly) and the topic being discussed. Annotation was performed via crowdsourcing, and annotation guidelines were carefully set to ensure high quality output, which was reflected in the high agreement levels for the different annotated features.

Experimental results confirm the importance of these features. For instance, including information about the topic and sentiment expression improves the performance of a baseline classifier by more than 10% absolute. Furthermore, results confirm the gap that exist between training and testing models on tweets from the same or from different topics. We also report a significant improvement of 13-14% when adding the *sentiment expression* feature, which suggests some dependency between sentiment polarity and how sentiment is expressed. It is worth mentioning that for these experiments, we used the manually-annotated features directly from the corpus, which is not a realistic scenario, just to highlight the potential benefits of using these features for sentiment analysis.

Future work include developing accurate machine learning models that leverage the existing annotation to perform both overall and target-based sentiment in Arabic tweets. It is also interesting, given tweets that are segregated by dialect and topic, to investigate cross-topic and cross-dialect solutions that will mitigate the amount of required resources that will be needed to perform sentiment analysis on any given piece of text.

Acknowledgment

This work was made possible by NPRP 6-716-1-138 grant from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

6. Bibliographical References

Abdul-Mageed, M. and Diab, M. T. (2012). Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In *LREC*, pages 3907–3914.

Abdul-Mageed, M., Diab, M. T., and Korayem, M. (2011). Subjectivity and sentiment analysis of modern standard arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 587–591. Association for Computational Linguistics.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Al-Kabi, M., Al-Ayyoub, M., Alsmadi, I., and Wahsheh, H. (2016). A prototype for a standard arabic sentiment analysis corpus. *Int. Arab J. Inf. Technol.*, 13(1A):163–170.

Al Sallab, A. A., Baly, R., Badaro, G., Hajj, H., El Hajj, W., and Shaban, K. B. (2015). Deep learning models for sentiment analysis in arabic. In *ANLP Workshop*, volume 9.

Al-Sallab, A., Baly, R., Hajj, H., Shaban, K. B., El-Hajj, W., and Badaro, G. (2017). Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):25.

Al-Smadi, M., Qawasmeh, O., Talafha, B., and Quwaider, M. (2015). Human annotated arabic dataset of book reviews for aspect based sentiment analysis. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pages 726–730. IEEE.

Al-Twairesh, N., Al-Khalifa, H., Al-Salman, A., and Al-Ohali, Y. (2017). Arasenti-tweet: A corpus for arabic sentiment analysis of saudi tweets. *Procedia Computer Science*, 117:63–72.

Aly, M. A. and Atiya, A. F. (2013). Labr: A large scale arabic book reviews dataset. In *ACL (2)*, pages 494–498.

Assiri, A., Emam, A., and Al-Dossari, H. (2016). Saudi twitter corpus for sentiment analysis. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(2):272–275.

Badaro, G., Baly, R., Hajj, H., Habash, N., and El-Hajj, W. (2014). A large scale arabic sentiment lexicon for arabic opinion mining. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 165–173.

Badaro, G., Baly, R., Akel, R., Fayad, L., Khairallah, J., Hajj, H., El-Hajj, W., and Shaban, K. B. (2015). A light lexicon-based mobile application for sentiment mining of arabic tweets. In *ANLP Workshop 2015*, page 18.

Baly, R., Hobeica, R., Hajj, H., El-Hajj, W., Shaban, K. B., and Al-Sallab, A. (2016). A meta-framework for modeling the human reading process in sentiment analysis. *ACM Transactions on Information Systems (TOIS)*, 35(1):7.

Baly, R., Badaro, G., El-Khoury, G., Moukalled, R., Aoun, R., Hajj, H., El-Hajj, W., Habash, N., and Shaban, K. B. (2017a). A characterization study of arabic twitter data with a benchmarking for state-of-the-art opinion mining models. *WANLP 2017 (co-located with EACL 2017)*, page 110.

Baly, R., Badaro, G., Hamdi, A., Moukalled, R., Aoun, R., El-Khoury, G., Al Sallab, A., Hajj, H., Habash, N., Shaban, K., et al. (2017b). Omam at semeval-2017 task 4: Evaluation of english state-of-the-art sentiment analysis models for arabic and a new topic-based model. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 603–610.

Baly, R., El-Khoury, G., Moukalled, R., Aoun, R., Hajj, H., Shaban, K. B., and El-Hajj, W. (2017c). Comparative evaluation of sentiment analysis methods across arabic dialects. *Procedia Computer Science*, 117:266–273.

Baly, R., Hajj, H., Habash, N., Shaban, K. B., and El-Hajj, W. (2017d). A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in arabic. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):23.

Biyani, P., Caragea, C., and Bhamidipati, N. (2015). Entity-specific sentiment classification of yahoo news comments. *arXiv preprint arXiv:1506.03775*.

Brody, S. and Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812. Association for Computational Linguistics.

Chen, X., Sun, Y., Athiwaratkun, B., Cardie, C., and Weinberger, K. (2016). Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*.

ElSahar, H. and El-Beltagy, S. R. (2015). Building large arabic multi-domain resources for sentiment analysis. In *CICLing (2)*, pages 23–34.

Farra, N. and McKeown, K. (2017). Smarties: Sentiment models for arabic target entities. *arXiv preprint arXiv:1701.03434*.

Farra, N., Challita, E., Assi, R. A., and Hajj, H. (2010). Sentence-level and document-level sentiment mining for arabic texts. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 1114–1119. IEEE.

Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 151–160. Association for Computational Linguistics.

Joty, S., Nakov, P., Màrquez, L., and Jaradat, I. (2017). Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 226–237.

Kirange, D. and Deshmukh, R. R. (2015). Emotion classification of restaurant and laptop review dataset: Semeval 2014 task 4. *International Journal of Computer Applications*, 113(6).

Li, Z. (2017). End-to-end adversarial memory network for cross-domain sentiment classification.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467.

Medhaffar, S., Bougares, F., Esteve, Y., and Hadrich-Belguith, L. (2017). Sentiment analysis of tunisian dialects: Linguistic ressources and experiments. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 55–61.

Mohammad, S. M., Salameh, M., and Kiritchenko, S. (2016). How translation alters sentiment. *J. Artif. Intell. Res.(JAIR)*, 55:95–130.

Mohit, B., Rozovskaya, A., Habash, N., Zaghouani, W., and Obeid, O. (2014). The first qalb shared task on automatic text correction for arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 39–47.

Nabil, M., Aly, M. A., and Atiya, A. F. (2015). Astd: Arabic sentiment tweets dataset. In *EMNLP*, pages 2515–2519.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. (2010). Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760. ACM.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Paolillo, J. C. and Das, A. (2006). Evaluating language

statistics: The ethnologue and beyond. *Contract report for UNESCO Institute for Statistics*.

Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.

Refaee, E. and Rieser, V. (2014a). An arabic twitter corpus for subjectivity and sentiment analysis. In *LREC*, pages 2268–2273.

Refaee, E. and Rieser, V. (2014b). Subjectivity and sentiment analysis of arabic twitter feeds with limited resources. In *Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop Programme*, page 16.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Rushdi-Saleh, M., Martín-Valdivia, M. T., Ureña-López, L. A., and Perea-Ortega, J. M. (2011). Oca: Opinion corpus for arabic. *Journal of the Association for Information Science and Technology*, 62(10):2045–2054.

Salameh, M., Mohammad, S., and Kiritchenko, S. (2015). Sentiment after translation: A case-study on arabic social media posts. In *HLT-NAACL*, pages 767–777.

Salem, F. (2017). Social media and the internet of things towards data-driven policymaking in the arab world: Potential, limits and concerns.

Sareah, F. (2015). Interesting statistics for the top 10 social media sites. *Small Business Trends*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Somasundaran, S. and Wiebe, J. (2009). Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 226–234. Association for Computational Linguistics.

Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, pages 1422–1432.

Wang, Y., Huang, M., Zhu, X., and Zhao, L. (2016). Attention-based lstm for aspect-level sentiment classification. In *EMNLP*, pages 606–615.

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

Building a Causation Annotated Corpus: The Salford Arabic Causal Bank - Proclitics

Jawad Sadek[1], Farid Meziane[2]

[1]National Institute for Health Research Innovation Observatory – Newcastle University
[2]School of Computing Science and Engineering – Salford University

jawad.sadek@newcastle.ac.uk, f.meziane@salford.ac.uk

Abstract

We introduce the Salford Arabic Causal Bank (SACB) corpus, a new corpus dedicated to Arabic *Causal* relations. Causality as a linguistic phenomenon can be expressed using different elements and grammatical expressions. In Arabic language, causal particles – *Purpose Lām*, *Causation Fa'a*, *Causation Ba'a*- are frequently prefixed to words; they play a key role in indicating causality. However, these particles give different meanings according to their position in the text. In fact, these meanings can be interpreted according to the context in which they occur. This ambiguity emphasizes the high demand for a large-scale corpus in which instances of these particles are annotated. In this paper, we present the first stage of building the SACB, which includes a collection of annotated sentences each of which contains an instance of a causal particle. The sentences were carefully examined by two specialist annotators to give an accurate account for each annotated instance. Arabic is a less–resourced language and we hope this corpus would help in building better Information Extraction systems.

Keywords: Arabic Annotated Corpus, Causal Relation, Information Extraction

1. Introduction

Automatic detection of *Causal* relations has gained popularity in the literature within different Natural Language Processing (NLP) applications such as Text Generation, in which causality is exploited to provide explanation and generate knowledge (Kaplan and Berry-Rogghe, 1991). Modern Information Retrieval researchers have focused on developing more efficient search engines by incorporating *Causal* relations into their lexical-based approach (Puente, 2011). Question Answering (QA) is another NLP field to which *Causal* relations is well suited. In particular, it plays a very major part in developing *why*-QA systems (Sadek and Meziane, 2016b; Azmi and Alshenaifi, 2014). Consider for example sentence (1) which contains a *Causal* relation holding between Units 1 and 2. We can return Unit 1 as a candidate answer for the question *"Why was Sarah late?"*

(1) [Because the car broke down,][1] [Sarah was late for school][2].

In Arabic, causality can be expressed using different linguistic elements and expressions. It can be classified into two major categories. The first one is **verbal causality**, which can be captured by the presence of nominal clauses for example, [المفعول لأجله (Accusatives of purpose)-المفعول المطلق (Cognate accusative)] or by causality connectors such as [لذا (therefore), بسبب (because) and من اجل (for)]. The second category is **context-based causality** that can be inferred by the reader using general knowledge without locating any of the previous indicators. This category includes various Arabic stylistic structures that express causality implicitly such as [الاستئناف (resumption), الشرط(condition), الاستثناء (exception)] (Haskour, 1990).

Within the first category, there is a significant group of inseparable particles that are always bound to words. We refer to this group as causal particles, or proclitics for short, and includes: *Purpose Lām* (لام التعليل) – *Causation Fa'a* (فاء السببية) and *Causation Ba'a* (باء السببية).

Arabic authors use these proclitics substantially to indicate causal meaning. In a previous study, we constructed a set of linguistic patterns to detect and extract *Causal* relations expressed in Arabic texts (Sadek and Meziane, 2016a). Several newspaper articles were surveyed in order to design three rule based algorithms that help in recognizing the cases in which the proclitics function as a causative conjunction. Our results reveal that combining the algorithms with the linguistic patterns model has boosted the efficiency by a large margin, improving the overall *recall* measure for Health and Science texts by 29% (out of 195 true positive *Causal* relations, 70 were indicated by proclitics). However, this improvement comes at the cost of *precision* which was reduced by 16% (out of 56 false positive *Causal* relations, 47 attributed to proclitics) i.e. 67% of relations returned by proclitics's algorithms were misclassified. This decline in precision highlights the ambiguity associated with these particles.

The Arabic language, so far, is under-resourced in terms of availability of knowledge base repositories. These resources play an important role in building robust NLP tools and support language technologies' researchers on developing and testing their solutions. Although there are a number of annotated corpora for Arabic, such resources are either 'low-level' (e.g. syntactical or morphological) annotated or they have been labelled with *Causal* relations while annotating other semantic relations. We argue that causation is a complex phenomenon and needs to have annotators to be trained and focus in particular on *Causal* relations.

The syntactical patterns of the Arabic *Causal* relations are rather complex and no general annotated corpus can

provide the diversity of *Causal* relations. So we cannot build on top of any pre annotated corpus but have to create a dedicated corpus of this type of relations. In the current work we introduce the first stage towards building the Salford Arabic Causal Bank (SACB). This stage has been conducted with the goal of collecting and annotating independent sentences where instances of proclitics occurred without regard for other causal indicators.

2. Data Collection

For the purpose of collecting our data, we used the untagged *arabiCorpus*[1] to gather all instances. It is a large corpus consisting of a variety of resources written in Modern Standard Arabic (MSA). The corpus has a Newspapers category containing approximately135 million words of articles published between 1996 and 2010 in different Arabic countries. This category is a good representative for real-world texts as it covers a wide variety of topics.

Searching the *arabiCorpus* for occurrences of words starting with *Lām*, *Fa'a* or *Ba'a*, (henceforth, target word) returns a huge number of matching instances. The issue here is that randomly sampling these instances yields an under-coverage dataset i.e. not every syntactical or semantic form is sufficiently included. This is inherited from the fact that proclitics tend to be highly skewed e.g. the vast majority occurrences of *Fa'a* in Arabic text do not express causation. In which case, most classifiers trained on such dataset would be biased toward major class.

In general, the collected instances must be independent and almost identically distributed. A carefully chosen sample is therefore vital in building a reasonably confident corpus that represents all proclitics' characteristics. To this end, we performed a multistage sampling. We first split the matching instances returned from initial searching (approximately 2.5 million instances) into separate groups according to the length of target words; words of the same length tend to share more linguistic characteristics e.g. grammatical category, morphological pattern. Splitting the data generated five clusters with target word's length of *n*= 2, 3, 4, 5, and over 5 letters; each cluster was then divided into different sub-groups that share one syntactical functionality.

Finally we performed a judgment sampling to avoid data bias. In this phase, the aim is to force the harvested instances to be reasonably balanced between causal and non-causal classes. We requested a native speaker to skim through all clusters and first to randomly select a number of instance that express causation and then to select equivalent number of instances that are non-causal. The number of instances drawn from each cluster was proportionate to the ambiguity of the cluster's population. For example, all instances belonging to clusters of two letters (e.g. لم – في - بث) are classified as non-causal, thus we can be confident that a small size of instances is sufficient to represent these clusters.

3. Annotation Scheme

We used GATE framework (Bontcheva et al., 2013) to support annotation tasks throughout all phases of building our corpus. GATE provides tools for adjudication, integrating multiple annotations set, running various NLP components and supports texts written in Arabic-like script orientation i.e. right-to-left. In addition it permits to create annotation schemas supported by W3C Schema which allows annotation types and features to be pre-specified. In this way, it facilitates the development of Gold Standards. The manual annotation phase was preceded by automatic pre-processing steps. All sentences passed through an NLP components pipeline comprising of the following processes: tokenization, sentence-splitting and POS tagging. We implement the last process using the Stanford POS tagger (Toutanova et al. 2003).

Before an annotation scheme and guidelines can be defined, it is necessary to make clear on what ground we make a judgment on whether the proclitic implies a causal function or not.

3.1 Causal Particles

Causal particles are one of the most complicated and ambiguous particles in Arabic language, as it express many different meaning (Wright et al. 1896). A brief explanation of the particles under consideration in this work is given here.

- **Lām:** It has a multifunctional role and many semantic properties insomuch that some grammarians count more than 30 different purposes of it. For example, (لام الجحود) *Lām of denial* as in "Kalid was not to drink milk" " لم يكن خالد ليشرب الحليب" and (لام الملك) *Lām of possession* when indicating the right of property, e.g. "Ahmad had a large car" " كان لأحمد سيارة كبيرة". However, our concern in this study is *Lām at-'taleel* (لام التعليل) or *Purpose Lām*, which indicates the purpose for which, or the reason why, a thing is done. In this context, the Arab grammarians take *Lām-at-'taleel* to function similarly to (لكي) or (لأن), for example, "he arose to help him" "قام لمعاونته".
- **Fa'a:** It may signal a consequential relationship between two elements or events occurring consecutively, as in "Khalid stood up, then Ahmad" "قام خالد فاحمد". *Fa'a* has also an adversative function, in which it expresses a contrast between two clauses, as in "He invited me, but I turned down his invitation" " دعاني فلم اجب دعوته". In addition, it has a role related to our study in which it contributes to indicating causation between two parts of a sentence, as in "He loved theatre so he excelled in it" "احب المسرح فابدع فيه" (Saeed and Fareh 2006).
- **Ba'a:** It also poses many difficulties. One use of this particle is "الظرفية" to express time and place, for example, "He travelled two days before me" "سافر قبلي بيومين". It can also be used to indicate adhesion "الإلصاق" e.g. "لان الدود يتعلق بالثمار" "because worms stick to the fruit". Another use is to form negation, as in "I don't Know" " لست بعالم". Moreover, it expresses the reason and cause, for example, "كان الاعتداء بقصد السرقة" "The attack committed with intent to steal".

[1] http://arabicorpus.byu.edu/index.php

3.2 Annotation Guidelines

The decision on whether a proclitic serves as a casual indicator may differ according to the way in which it is perceived e.g. syntactic or semantic. In other words, a proclitic which appear to be grammatically a causal particle, the causality may not be contextually perceivable. Since we are dealing with causation from a discourse perspective, we embrace the following principles: *Causal* relation occurs between an event (*the cause*) and a second event (*the effect*) in which the second event is understood as a consequence of the first. When deciding whether there is a *Causal* relation, the annotators were advised to ask whether event B (*effect*) would have occurred if event A (*cause*) had not occurred. If A is a sufficient though not a necessary condition for B to occur, we conclude that A caused B.

A related issue is whether a *cause* or *effect* can be a fact, or whether they have to be an event. In this work, we don't limit *cause* or *effect* to particular types of entities. Thus, an *effect* can be an event, a fact, a method; a *cause* can refer to a reason, motivation, human action, psychological, technological causation etc. We advised annotators to include all the various types. In this context, we label sentences (2) and (3) as two instances holding *Causal* relations indicated by *Ba'a* where the underlined metaphor in sentence (2) represents the *effect*, while the method the *woman* embrace in sentence (3) constitutes the *effect* part of the relation.

(2) نحاول التستر على ضعفنا باخفاء رؤوسنا في التراب، كالنعامة.

"We are trying to cover up our weakness by burying our head in the sand like ostriches."

(3) يتحدث النص عن عجوز تصبر نفسها على الانتظار باسترجاع الذكريات السعيدة من حياتها.

"The text is about an old woman who passes her time waiting by remembering happy moments in her life."

Taking these assumptions into account, the annotators were required to read the entire sentence so that they can make reliable interpretations to the writer's purpose. Then to decide whether the target word indicates a causation based on two facts: both *cause* and *effec*t arguments are securely presented in the sentence where the *effect* has to be explicitly the result of the *cause*; plus each argument constitutes an independent clause i.e. they don't overlap. For example, we classify the particle *Fa'a* in sentence (4) as non-causal. The text does not reveal the fact that made the writer reach his conclusion; and there is no referring expression to any idea mentioned in the previous sentences. As such the reason is only vaguely specified.

(4) لقد قرأت ذات يوم كتابا يقول «كيف تصبح مليونيرا» فلما انتهيت منه ادركت انني لن اصبح مليونيرا.

"I once read a book titled How to Become a Millionaire and when I finished it, I realized that I would never become a millionaire."

It is worth noting that even if the target word indicates causation, the first letter could be a basic unit of the word i.e. it is not a proclitic. The annotators need to be aware of this and should not be tempted to assign a causal status. For example, the target word 'بناء' "at" in sentence (5) starts with *ba'a* that is a part of its original root. The *cause* and *effect* arguments were also annotated if the target word was classified as causal.

(5) التحقت بكلية الحقوق بناء على رغبة امي، فقد ارادت لي ان اصبح محامي مثل والدي.

"I enrolled in the law school at my mother's wish as she wanted me to become a lawyer to follow my father."

Next, the annotators consider a window of five words surrounding the target word and override all POS annotations in this window with new fine-grained ones. This entails assigning different POS tags on sub-word level. The rule-based approach indicates that prefixes and suffixes of surrounding words provide useful hints on proclitics' functionality (Sadek and Meziane, 2016a). All instances annotated according to Stanford POS tag-set, however, we expanded this set so it becomes appropriate to perform fine-grained tagging. For example, we added TIM (ظرف زمان) "adverb of time" - LOC (ظرف مكان) "adverb of place" - PRPY (ضمير متصل) "inseparable pronoun".

The annotators were also required to assign an annotation label referring to the "الوزن الصرفي" "morphological pattern" of the target word. The majority of Arabic words are derived by applying a set of morphological patterns to consonantal roots to which affixes and infixes are added. Morphological patterns are abstractions which can be considered as an indicator of the common concept of the meaning of the word such as *tool* an *event place*/*time* and *instrument*. This classification constitutes a valuable feature in recognizing the role of certain proclitic. For example, a proclitic can be classified as non-causal if the target word belongs to a set of nominal patterns e.g. اسم الفاعل '*present participle*', اسم مكان '*noun of place*'.

3.3 Annotation Process and Adjudication

Two native speakers of Arabic were engaged in the manual annotation process. One annotator (identified as annotator A) was a graduate student in the faculty of Arabic literature. The second annotator (identified as annotator B) was a teaching assistant who has been educated entirely in Arabic. Annotators were trained using the GATE tool on a training set of examples randomly selected from the original dataset. They were asked to identify the function of each proclitic in the training set, and their judgments were compared with the function we had identified in the sentences. We then discussed with each annotator the instances where their judgments differed from ours and clarified the guidelines.

However, it is inevitable that the annotators disagree about the function of some proclitics. In fact, the topic of causation is a matter of debate among experts belonging to this filed (Davidson, 1980; Mackie, 1980). For example, examining the function of the target word "بالتفرج" "looking" in sentence (6), we observed that annotator B assigned causal status to the event " التفرج على لوحاتي " "looking at my drawings", considering the *effect* argument is "keep busy". Annotator A on the other hand conceived the aforementioned event as a request.

In order to create a gold standard set of annotations, we automatically correct all minor mistakes made by annotators using a script written in Groovy language. These corrections are not to interfere or change

annotators' decision, but rather to fix inconsistency e.g. word's length, letter-spacing. We reconciled the differences between annotators by first accepting only instances where both annotators agreed on the binary decision on whether a proclitic indicates causation. Thus we eliminated approximately 300 instances. Then we examined the consensus set for differences in the POS tags. In case there was any disagreement, we included the ones annotated by annotator A as she is an Arabic literature specialist. Table 1 summarizes the main aspects of the final annotated instances: number of instances (N), number of annotated text units (Tokens), number of instances assigned the causal class (causal), number of instances assigned the non-causal class (¬causal). Table 2 illustrates the statistics of instances over the five main clusters. Gate annotation tool format documents in GATE XML style. We converted the documents using another Groovy script so that all annotated instances are encoded in a lightweight XML. Figure 1 provides an excerpt of one instance.

(6) اشغل نفسك بالتفرج على لوحاتي حتى أعد فنجان قهوة وارجع اليك.

"Keep yourself busy looking at my drawings until I make a cup of coffee and come back."

Proclitic	N	Tokens	causal	¬causal
Lām	984	31564	439	545
Fa'a	577	20097	247	330
Ba'a	601	17912	290	311
Total	2162	69573	976	1186

Table 1: Statistics of the dataset

Proclitic	2	3	4	5	+5
Lām	17	61	230	234	442
Fa'a	9	81	111	184	192
Ba'a	22	27	100	114	338

Table 2: Statistics of the dataset based on proclitic's length

4. Related Work

Some research works for Arabic focused on developing annotated corpus with discourse relations. The Arabic Discourse Treebank was generated by (Al-Saif and Markert, 2011) based on the Arabic Penn Treebank. They collected a list of 80 explicit discourse connectives to recognize 18 discourse relations that link adjacent discourse units (DU). The relations are subclasses of four main classes: Temporal, Contingency, Comparison and Expansion. This corpus contains approximately 600 sentences annotated with *Causal* relations under the Contingency class. Another attempt presented by (Keskes et al., 2014) to identify implicit and explicit discourse relations. The authors created an annotated corpus on top of a set of documents extracted from the Discourse Arabic Treebank (Maamouri et al., 2016). The annotation process was performed according to the principles of the Segmented Discourse Representation Theory. They employed the Maximum Entropy model to automatically identify 24 discourse relations holding between adjacent and non-adjacent DUs. The relations were grouped into four top levels classes: Thematic, Temporal, Structural and Causal; of which there are 158 instances annotated with the cause-effect category.

5. Conclusion

There is a lot of uncertainty surrounding the decision about when two events are causally linked. However, the importance and difficulty of extracting causal information suggest that additional efforts are needed in order to reliably create mature language resources. In Arabic, *Causal* relations indicated by causal particles account for a high percentage of the total *Causal* relation in texts. In the current research we created a causation corpus annotated with instances containing words prefixed with certain proclitic along with *cause* and *effect* arguments. In future, we will extend the corpus to include other causal indicators.

```
<Sentence Id="0309" Start="0" End="95">
 <Text>. طلب من اسماعيل ان يأتيه بحجر يكون علامة للناس فذهب اسماعيل يبحث
  عن حجر يؤدي هذا الغرض</Text>
  <Annotations>
   <Annotation Id="11347" Type="Target Word" Start="55" End="58">
     <Features>
       <Length>4</Length>
        <Template>ففعل</Template>
       <Status >causal</Status>
        <String>فذهب</String>
       <Kind>Fa'a</Kind>
     </Features>
   <Annotation Id="11348" Type="Argument" Start="0" End="53">
     <Features>
       <Length>53</Length>
        <String> طلب من اسماعيل ان يأتيه بحجر يكون علامة للناس
</String>
       <Kind>cause</Kind>
     </Features>
  </Annotation>
   <Annotation Id="11349" Type="Argument" Start="58" End="95">
     <Features>
       <Length>38</Length>
        <String> ذهب اسماعيل يبحث عن حجر يؤدي هذا الغرض</String>
       <Kind>effect</Kind>
     </Features>
  </Annotation>
  <Annotation Id="11350" Type="Token" Start="0" End="2">
     <Features>
        <String>طلب</String>
       <Type>arabic</Type>
       <Kind>word</Kind>
       <Length>3</Length>
       <Category>VBD</Category>
     </Features>
   </Annotation>
                  .
                  .
                  .
```

Figure 1: Excerpt from the Salford Arabic Causal Bank.

Acknowledgements

This research work has been funded by Salford University.

6. References

Al-Saif, A., and Markert, K. (2011). Modelling discourse relations for Arabic. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, (EMNLP 2011), Edinburgh.

Azmi, A., and AlShenaifi, N. (2014). Handling "why" questions in Arabic. *In The 5th International Conference on Arabic Language Processing (CITALA'14).*

Bontcheva, K., Cunningham, H., Roberts, I., Roberts, A., Tablan, V., Aswani, N., and Gorrell. G. (2013). Gate Teamware: A Web-based, Collaborative Text Annotation Framework. Language Resources and Evaluation. Volume 47, Issue 4.

Davidson, D. (1980). Causal relations. Essays on actions and events. Oxford University Press. pp. 149-162

Haskour, N. (1990). Al-Sababieh fe tarkeb Al-Jumlah Al-Arabih. Master's thesis, Aleppo University, Aleppo, Syria.

Kaplan, R and Berry-Rogghe, G. (1991). Knowledge-based acquisition of causal relationships in text. Knowledge Acquisition 3, pp 317–337.

Keskes, I., Zitoune, F.B., and Belguith, L. H. (2014). Learning explicit and implicit arabic discourse relations. Journal of King Saud University, computer and Information Sciences vol: 26 (4), PP 398–416.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North America Chapter of the Association for Computational Linguistics on Human Language Technology*, Volume 1. 173-180.

Maamouri, M., Bies, A., Kulick, S. Krouma, S., Gaddeche, Zaghouani,W., 2010b. Arabic Treebank (ATB): Part 3 Version 3.2. Linguistic Data Consortium, Catalog No.: LDC2010T08.

Mackie, J.L. (1980). The cement of the universe: A study of causation. Oxford University Press.

Puente, C., Sobrino, A., and Olivas, J.A. (2011). Retrieving crisp and imperfect causal sentences in texts: From single causal sentences to mechanisms. In *Soft Computing in Humanities and Social Sciences*. pp 175–194.

Sadek, J., and Meziane, F. (2016a). Extracting Arabic causal relations using linguistic patterns. Journal ACM Translations on Asian and Low-Resource Language Information Processing 15, 3, Article 14.

Sadek, J., and Meziane, F. (2016b). A discourse-based approach for Arabic question answering. Journal ACM Translations on Asian and Low-Resource Language Information Processing 16, 2, Article 11.

Saeed, A., and Fareh, F. (2006). Difficulties encountered by bilingual Arab learner in translating Arabic "fa" into English. The International Journal of Bilingual Education and Bilingualism 9, 1, pp 19–32.

Wright, W., and Caspari, C. (1896). A grammar of the Arabic language. Cambridge University Press, Cambridge, England.

Dial2MSA: A Tweets Corpus for Converting Dialectal Arabic to Modern Standard Arabic

Hamdy Mubarak
QCRI, Hamad Bin Khalifa University (HBKU), Doha, Qatar
hmubarak@hbku.edu.qa

Abstract

Modern Standard Arabic (MSA) is the official language used in formal communications while Dialectal Arabic (DA) refers to the spoken languages in different Arab countries and regions, and they are widely used on social media for daily communications. There are differences between DA and MSA at almost all levels, and resources for DA are very limited compared to MSA. In this paper, we present Dial2MSA corpus; the first and largest corpus of dialectal tweets with translations to MSA as provided by large number of native speakers through crowdsourcing. We describe how we collected the tweets, annotated them and measured translation quality. We aim that Dial2MSA can promote researches in understanding and quantifying differences between DA and MSA, dialect identification, converting DA to MSA (hence using MSA resources) and machine translation (MT) among other applications. Roughly, the corpus contains 5,500 and 5,000 tweets written in Egyptian and Maghrebi dialects with verified MSA translations (16,000 and 8,000 pairs in order), and 6,000 tweets written in Levantine and Gulf dialects with MSA translations (18,000 pairs for each without verification). The corpus is freely available for research purposes.

Keywords: Arabic Dialects, Dialect to MSA conversion, Parallel Corpus, Crowdsourcing

1. Introduction

Modern Standard Arabic (MSA) is the lingua franca of the Arab world, and it's used in official communications and speeches such as books, educational materials and newspapers. On the other hand, Dialectal Arabic (DA) refers to local dialects (or languages) spoken in different countries and regions, and they differ from country to another and sometime from city to another in vocabulary, morphology, and spelling among other things. These dialects are widely used on daily interactions and on social media platforms such as Facebook and Twitter.

Conventionally, researchers in the Arabic Natural Language Processing (NLP) field divide DA into major dialectal groups, namely: Egyptian (EGY), Maghrebi (MGR) spoken in the Maghreb region or North Africa, Levantine (LEV) spoken in the Levant, Gulf (GLF) spoken in the Arabic Peninsula, and Iraqi (IRQ). Sometimes IRQ is considers as one of the Gulf dialects.

There are many resources for MSA, such as large annotated corpora and tools, for different NLP tasks (e.g morphological analysis, parsing, machine translation, etc.) which generally achieve high scores. Compared to MSA, DA suffers from lack of resources. One possible solution for some tasks is to convert DA to MSA (i.e. use MSA as a pivot language or a bridge) such as researches done by (Bakr et al., 2008), (Al-Gaphari and Al-Yadoumi, 2010), (Sawaf, 2010), (Sajjad et al., 2013), (Salloum and Habash, 2013) and (Shaalan, 2016) to enhance translating DA to English.

Moreover, there is a lot of work in the MT field to convert from a resource-poor language to other languages by pivoting on a closely-related resource-rich language such as in (Durrani et al., 2010), (Hajič et al., 2000), and (Nakov and Tiedemann, 2012). This conversion can be done at different levels: character level transformation, word level translation or language-specific rules.

Dialect to MSA conversion or translation is usually performed using handcrafted rules and heuristics that require deep linguistic knowledge and extensive manual efforts. As reported by (Sajjad et al., 2013), conversion can also be done using translation methods but generally this requires parallel data (pairs of DA and MSA) which is not available. They manually created a lookup table of EGY-MSA words, and applied an automatic character-level transformation model to change EGY to something similar to MSA, and this gave a gain of 1.87 BLEU points for translating EGY to English.

In this paper, we introduce Dial2MSA; a new large-scale corpus of DA-MSA pairs of tweets for major dialects (EGY, MGR, LEV and GLF) as written by native speakers. We aim to support the field of dialectal NLP and reduce the effort of building linguistic rules for conversion by providing parallel data that can be used by statistical machine translation (SMT) techniques between these closely-related languages.

It is worth mentioning that Dial2MSA is different than the Arabic multi-dialectal parallel corpus published by (Bouamor et al., 2014) in different aspects:

- Bouamor's corpus contains translations of 2,000 EGY sentences to Palestinian, Syrian, Jordanian and Tunisian dialects in addition to MSA. Starting from EGY can be considered as biased input, and does not give the variety and naturalness found in native tweets written in these dialects.
- Each sentence in Bouamor's corpus is translated by only one person (the same person) per dialect, and in our corpus hundreds of native speakers participated in the translation process (multiple translations for each tweet) which guarantees wide range of opinions.

- Our corpus size is bigger.

Next sections have details about corpus collection, annotation and measuring translation quality. Then some statistics and examples are provided.

2. Data Collection

From a corpus of 175M Arabic tweets collected during March 2014[1], we filtered tweets using very strong dialectal words for each major dialect to extract dialectal tweets. These dialectal words (140 words) are mostly function words that are used exclusively in each dialect and they were revised by native speakers. Initial list was obtained from (Mubarak and Darwish, 2014b) then it was revised manually for better quality. Examples are shown in Table 1 and the full list can be downloaded from `http://alt.qcri.org/~hmubarak/EGY-MGR-LEV-GLF-StrongWords.zip`.

Dialect	Examples of dialectal words
EGY	ده ، عاوز ، إزاي
	this, want, how
MGR	بزاف ، علاش ، هكي
	very much, why, like this
LEV	هيك ، مشان ، عنجد
	like this, for, really
GLF	إشلون ، شصار ، مو بطبيعي
	how, what happened, not natural

Table 1: Strong dialectal words

For each dialect, we removed duplicate tweets, and selected tweets having lengths between 25 and 90 Arabic characters without counting mentions, URL's, etc. (roughly between 5 and 15 words), then selected random 6,000 tweets for the next annotation process.

3. Data Annotation

We created annotation jobs (Task1), one for each dialect, on CrowdFlower[2] (CF) where we showed dialectal tweets to annotators and asked them to provide corresponding MSA translations or conversions to have pairs of DA-MSA. Annotators were selected from the the countries that speak the target dialect (e.g. for MGR, annotators are restricted to be from Maghreb countries).

For quality control, we used the code and applicable best practices suggested by (Wray et al., 2015) and (Mubarak, 2017b) to prevent, as much as possible, bad annotations for different types of poor translation. Each dialectal tweet was converted to MSA by different annotators (5 for EGY and 3 for other dialects), and around 200 annotators contributed in each annotation task. This gives a wide diversity of opinions needed for such tasks. Figure 1 shows a sample EGY tweet and its MSA translations as provided by different annotators.

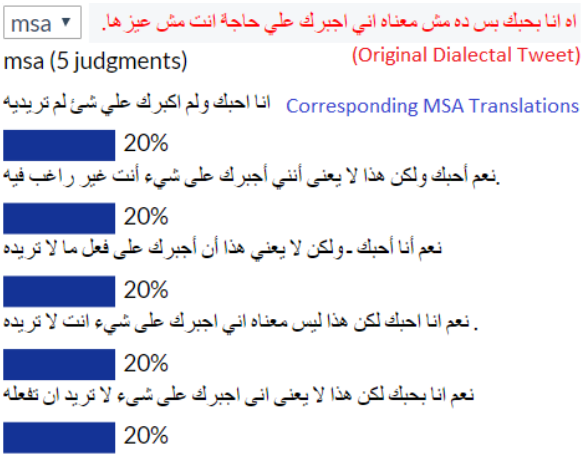

Figure 1: CF Task1: Converting DA to MSA

Quality of annotation at CF can be increased by using test questions where we provide their correct or gold answers, and annotators must pass a minimum threshold (typically 70%) of these test questions to continue. But because CF has limited capabilities in text comparison, and sentences can be expressed in many different ways, it's hard to list all possible forms of MSA sentences that can be used as gold answers to test questions. So to increase quality of the provided DA-MSA pairs, we created another annotation job for each dialect (Task2) to verify whether each pair is correct (i.e. having same meaning) or not. In this task, quality was controlled by using 50 test questions (correct pairs), and annotators should pass successfully a threshold of 80% to consider their work. Each pair was judged by 3 annotators who speak the target dialect. Sample annotation is shown in Figure 2.

Figure 2: CF Task2: Verify DA-MSA pairs

Translation jobs were completed for all dialects, and verification jobs of of the collected pairs were launched and completed for EGY and MGR because there are many annotators from theses regions (33% and 30% in order as obtained from recent surveys for Arab annotators on CF (Mubarak and Darwish, 2016)). We plan to verify collected pairs for other dialects as well.

[1]Using Twitter API (`http://dev.twitter.com`) with language filter assigned to "lang:ar"

[2]Crowdsourcing platform: `www.crowdflower.com`

Figures 3, 4, 5, and 6 show examples of dialectal tweets for each dialect and their MSA translations as provided by annotators. Dialectal words and their equivalent MSA words are marked with different colors.

EGY to MSA	EGY
ليس هناك احسن من الافراد الذين يحفظون السر	مفيش احلى من الناس اللى بتحفظ السر
ليس هناك اجمل من الاشخاص الذين يحفظون السر	مفيش احلى من الناس اللى بتحفظ السر
لا يوجد أحلى من الناس التي تحتفظ بالسر	مفيش احلى من الناس اللى بتحفظ السر

Figure 3: Example of EGY to MSA conversion

MGR to MSA	MGR
أنا لم أعد اتحمل أكثر من هذا	انا معش حا نتحمل اكثر من هكي
انا لن اتحمل اكثر من هكذا	انا معش حا نتحمل اكثر من هكي
انا لناتحمل اكثر من هذا لا استطيع	انا معش حا نتحمل اكثر من هكي

Figure 4: Example of MGR to MSA conversion

LEV to MSA	LEV
الهي لا تحرمني من هكذا اصدقاء	الهي ما انحرم من هيك صحاب
اللهم لا تحرمني من هكذا أصحاب	الهي ما انحرم من هيك صحاب
يا رب لا تحرمني من هؤلاء الاصدقاء	الهي ما انحرم من هيك صحاب

Figure 5: Example of LEV to MSA conversion

4. Data Quality

To get a rough estimate about the quality of obtained translations, we randomly selected 100 EGY tweets and their verified MSA translations (410 sentences), and asked a professional linguist to do needed corrections to make MSA sentences free of spelling and grammar errors, and retain the whole meaning of original tweets[3].

For comparison, we normalized MSA translations, before and after linguistic revision, to solve common spelling mistakes in some letters. For example, we converted all shapes of Hamza to plain Alif, Alif Maqsoura to dotted Yaa, and Taa Marbouta to Haa (الأخطاء الشائعة في الهمزات والتاء المربوطة والألف المقصورة), and removed punctuation marks. Correcting such errors is fairly easy and can achieve high accuracy by consulting a large clean corpus such as of `Aljazeera.net` as shown in (Mubarak and Darwish, 2014a). The overlap between translations before and after linguistic revision was 90% indicating high annotation quality obtained from non-experts.

[3]Linguistic corrections can be downloaded from: `http://alt.qcri.org/~hmubarak/EGY2MSA-sample-correction.zip`

GLF to MSA	GLF
الذي يريدك يعرف كيف يحافظ عليك	إلي يبيك...يعرف اشلون يحافظ عليك
الشخص الذي يريدك يعرف جيدا كيف يحافظ عليك	إلي يبيك...يعرف اشلون يحافظ عليك
من يريدك يعرف كيف يحافظ عليك	إلي يبيك...يعرف اشلون يحافظ عليك

Figure 6: Example of GLF to MSA conversion

Figure 7 shows examples of MSA translations obtained from CF, and their corrections for the EGY tweet:
احساس حلو اوي لما تلاقي حد بيقولك أنا بفرح اوي لما بتكلم معاك.
Spelling and grammar errors and their corrections are marked in different colors. Most errors are common and can be recovered, and there are some grammatical errors (case ending) and few split/merge errors. We estimate MSA translations for other dialects to have similar accuracy and they all need spelling correction.

We noticed that some translations are a bit unnatural, and this can be checked probably by using language models trained on MSA. We leave this for future work.

For tweets having multiple translations, if we want to get the best translation with minimum errors, we can use ROVER algorithm to combine these translations. ROVER (Recognizer output voting error reduction) (Fiscus, 1997) is used in automatic speech recognition to implement a "voting" or rescoring process for combining outputs of multiple speech recognizers (translations in our case). It seeks to reduce word error rates by exploiting differences in the nature of the errors in multiple outputs as shown in Figure 8.

5. Preliminary Data Analysis

Statistics about Dial2MSA corpus are listed in Table 2 and it can be downloaded from `http://alt.qcri.org/~hmubarak/EGY-MGR-LEV-GLF-2-MSA.zip`.

We started by 6,000 tweets for each dialect, and approximately for EGY and MGR, we obtained 5,500 and 5,000 tweets[4] with 16,000 and 8,000 verified MSA translations respectively, i.e. almost half the annotations of Task1 were approved in Task2. For LEV and GLF, we have 18,000 MSA translations per each and they need verification.

For the verified DA-MSA pairs for EGY and MGR, we calculated number of words, average number of words per sentence, and the Overlap Coefficient (OC) (#common words in DA and MSA / minimum length) as suggested by (Bouamor et al., 2014) for normalized words. Results are shown in Table 3. Their OC values for Egyptian and Tunisian dialects are 0.45 and 0.31 in order.

6. Resource Description and Benefits

In this paper, we created Dial2MSA; a corpus of parallel pairs of DA tweets and their conversions or translations to

[4] All translations of some tweets were rejected

EGY to MSA (Linguistic Revision)	EGY to MSA (CrowdFlower)
إحساس جميل جدا حين تجد من يقول لك : أنا سعيد جدا حين أتحدث إليك	احساس جميل جدا حين تجد من يقول لك انا سعيد جدا حين أتحدث اليك
إحساس جميل حينما أجد أحدًا يقول لي : أنا سعيد حينما أتحدث معك	احساس جميل حينما اجد حد يقول لي انا سعيد حينما اتحدث معك
إحساس جميل عندما : شخص يخبرك أنه يسعد جدا عندما تتحدث معه	احساس جميل عندما شخص يخبرك ان يسعد جدا عندما تتحدث معه
إنه إحساس جميل عندما تجد شخصًا يخبرك بأنه فرح عند التحدث معك	انه احساس جميل عندما تجد شخص يخبرك بانه فرح عند التحدث معك
شعور جميل جدا أن تجد من يقول لك : أنا أفرح جدا بحديثي معك	شعورجميل جدا ان تجد من يقول لك انا افرح جدا بحديثى معك

Figure 7: Linguistic revision example of MSA translations

Dialect	#Orginial Tweets	#MSA (Task1)	#Verified MSA (Task2)	%	#Tweets having MSA	Average #MSA/Tweet
EGY	6,000	30,000	16,355	55%	5,565	2.94
MGR	6,000	18,000	7,912	44%	4,953	1.6
LEV	6,000	18,000	-	-	-	-
GLF	6,000	18,000	-	-	-	-

Table 2: Statistics about Dial2MSA corpus

there's	a	lot	of	@	like	societies	@	@	ruin	engineers	and	lakes
there's	the	labs	@	@	like	societies	@	for	women	engineers	i	think
there's	the	last	@	@	like	societies	@	true	of	engineers	and	like
was	@	alive	@	the	legal	society	is	for	women	engineers	and	like
there's	a	lot	of	@	like	society's	@	@	through	engineers	@	like

REF: there's a lot OF like societies for women engineers and like
HYP: there's a lot ** like societies for women engineers and like

Figure 8: Aligning different outputs using ROVER

MSA as obtained from native speakers. We used crowdsourcing platform with quality control settings applied at different levels to have high quality of annotations with a wide variety of opinions which is normally not available in traditional companies. The cost of annotation jobs is less expensive and progress is fast compared to normal workers, and quality is comparable to language experts.

The obtained parallel DA-MSA pairs can help in understanding and quantifying similarities and differences between DA and MSA at different levels (phonology, morphology, and syntax), and enhancing dialectal Arabic NLP. Conversion was applied at sentence level (i.e. context is considered) which gives high accuracy.

Mapping between DA and MSA at different levels (characters, words or patterns) can be obtained automatically with high accuracy using alignment techniques because in most cases, there are no much differences in word order between them. This reduces the need for writing linguistic rules for DA to MSA conversion which requires a lot of experience and effort. For example, we can use Smith-Waterman algorithm[5] to align dialectal words and MSA counterparts with high accuracy as shown in Figure 9.

MSA words can also be used as pivots to align dialectal words in different dialects having the same meaning, ex: نحن = إحنا، نحنا، حنا.. (writing variations of "we" in MSA and DA).

EGY	MSA	MGR	MSA
معظم	معظم	علي	على
النصايح	النصائح	فكرا	فكرة
اللى	التي	تتنرفزو	هل
بننصح	ننصح		تغضبون
بيها	بها	مني	مني
غيرنا	غيرنا	لما	عندما
احنا	نحن	نسال	اسأل
مبنعملهاش	لا	هذي	هذه
	نفعلها	شن	ما
			هو
		معناها	معناها
		وهكي	وهكذا

Figure 9: Examples of aligning DA and MSA

7. Conclusion and Future Work

In this paper, we presented Dial2MSA; a corpus of DA tweets and their translations to MSA. This is the first and largest corpus available for DA to MSA conversion where original raw tweets are written by native speakers for each dialect which gives the needed naturalness and diversity found on social media sites.

[5] https://en.wikipedia.org/wiki/Smith-Waterman_algorithm

Dialect	#Words (Tweets)	#Words (MSA)	#Unique Words (Tweets)	#Unique Words (MSA)	#Words/sentence (Tweets)	#Words/sentence (MSA)	**Overlap Coeff.**
EGY	77,800	206,989	17,399	31,288	13.9	12.6	**0.33**
MGR	53,351	85,557	18,856	19,908	10.7	10.8	**0.38**

Table 3: Statistics about verified DA-MSA translation pairs

Translations of tweets are provided by native speakers through crowdsourcing, and each tweet is translated (and translations are verified) by different annotators to have variety of opinions. We measured quality of samples from the obtained pairs and showed that it's comparable to quality of language experts. The corpus is freely available for research purposes.

We plan to study the usefulness of this corpus on automatic translation of DA to MSA, translation across dialects, and from DA to English through pivoting on MSA. Also, we plan to correct spelling and grammar mistakes in the annotations and revise the automatic alignment to have more accurate and rich data.

It's worth mentioning that in (Mubarak, 2017a), translating EGY to MSA was applied at word level (i.e. lookup table) without having translations of complete tweets. For example, the word بس was translated to فقط، لكن (only, but). We estimate that translating complete tweets (such as in Dial2MSA corpus) would be more useful, and can produce a more fluent translation to MSA, and therefore better translation to English for example. Besides, using alignment algorithms can extract entries in DA-MSA lookup tables accurately especially for common words. Benefits of using translations of complete tweets over (or maybe with) individual words need to be experimented.

8. Bibliographical References

Al-Gaphari, G. and Al-Yadoumi, M. (2010). A method to convert sana'ani accent to modern standard arabic. *International Journal of Information Science & Management*, 8(1).

Bakr, H. A., Shaalan, K., and Ziedan, I. (2008). A hybrid approach for converting written egyptian colloquial dialect into diacritized arabic. In *The 6th international conference on informatics and systems, infos2008. Cairo university*.

Bouamor, H., Habash, N., and Oflazer, K. (2014). A multidialectal parallel corpus of arabic. In *LREC*, pages 1240–1245.

Durrani, N., Sajjad, H., Fraser, A., and Schmid, H. (2010). Hindi-to-urdu machine translation through transliteration. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 465–474. Association for Computational Linguistics.

Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover). In *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*, pages 347–354. IEEE.

Hajič, J., Hric, J., and Kuboň, V. (2000). Machine translation of very close languages. In *Proceedings of the sixth conference on Applied natural language processing*, pages 7–12. Association for Computational Linguistics.

Mubarak, H. and Darwish, K. (2014a). Automatic correction of arabic text: a cascaded approach. *ANLP 2014*, page 132.

Mubarak, H. and Darwish, K. (2014b). Using twitter to collect a multi-dialectal corpus of arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 1–7.

Mubarak, H. and Darwish, K. (2016). Demographic surveys of arab annotators on crowdflower. In *Weaving Relations of Trust in Crowd Work: Transparency and Reputation across Platforms Workshop (WebSci16)*.

Mubarak, H. (2017a). Analysis and quantitative study of egyptian dialect on twitter. In *The 3rd International Workshop on Natural Language Processing for Informal Text (NLPIT 2017), Maastricht, Italy*.

Mubarak, H. (2017b). Crowdsourcing speech and language data for resource-poor languages. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017. AISI 2017, Advances in Intelligent Systems and Computing, vol 639. Springer, Cham*.

Nakov, P. and Tiedemann, J. (2012). Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 301–305. Association for Computational Linguistics.

Sajjad, H., Darwish, K., and Belinkov, Y. (2013). Translating dialectal arabic to english. In *ACL (2)*, pages 1–6.

Salloum, W. and Habash, N. (2013). Dialectal arabic to english machine translation: Pivoting through modern standard arabic. In *HLT-NAACL*, pages 348–358.

Sawaf, H. (2010). Arabic dialect handling in hybrid machine translation. In *Proceedings of the conference of the association for machine translation in the americas (amta), denver, colorado*.

Shaalan, K., B. H. Z. I. (2016). Transferring egyptian colloquial dialect into modern standard arabic. In *International Conference on Recent Advances in Natural Language Processing (RANLP 2007), John Benjamins 2016*.

Wray, S., Mubarak, H., and Ali, A. (2015). Best practices for crowdsourcing dialectal arabic speech transcription. In *ANLP Workshop 2015*, page 99.

Creating an Arabic Dialect Text Corpus by Exploring Twitter, Facebook, and Online Newspapers

Areej Alshutayri[1,2] **and Eric Atwell**[1]
[1]School of Computing
University of Leeds, LS2 9JT, UK
{ml14aooa, E.S.Atwell}@leeds.ac.uk

[2]Faculty of Computing and Information Technology
King Abdul Aziz University, Jeddah, Saudi Arabia
aalshetary@kau.edu.sa

Abstract

In the last several years, the research on Natural Language Processing (NLP) on Arabic Language has garnered significant attention. Almost all Arabic text is in Modern Standard Arabic (MSA) because Arab people are writing in MSA at all formal situations, except in informal situations such as social media. Social Media is a particularly good resource to collect Arabic dialect text for NLP research. The lack of Arabic dialect corpora in comparison with what is available in dialects of English and other languages, showed the need to create dialect corpora for use in Arabic dialect processing. The objective of this work is to build an Arabic dialect text corpus using Twitter, and Online comments from newspaper and Facebook. Then, create an approach to crowdsourcing corpus and annotate the text with correct dialect tags before any NLP step. The task of annotation was developed as an online game, where players can test their dialect classification skills and get a score of their knowledge. We collected 200K tweets, 10K comments from newspaper, and 2M comments from Facebook with the total words equal to 13.8M words from five groups of Arabic dialects Gulf, Iraqi, Egyptian, Levantine, and North African. This annotation approach has so far achieved a 24K annotated documents; 16K tagged as a dialect and 8K as MSA, with the total number of tokens equal to 587K. This paper explores Twitter, Facebook, and Online newspaper as a source of Arabic dialect text, and describes the methods were used to extract tweets and comments then classify them into groups of dialects according to the geographic location of the sender and the country of the newspaper, and Facebook page. In addition to description of the annotation approach which we used to tag every tweet and comment.

Keywords: Arabic Dialects, Annotation, Corpus, Crowdsourcing

1. Introduction

The Arabic language consists of multiple variants, some formal and some informal (Habash, 2010).
The formal variant is Modern Standard Arabic (MSA). The MSA is understood by almost all people in the Arab world. It is based on Classical Arabic, which is the language of the Qur'an, the Holy Book of Islam. MSA used in media, newspaper, culture, and education; additionally, most of the Automatic Speech Recognition (ASR) and Language Identification (LID) systems are based on MSA. The informal variant is Dialectal Arabic (DA). It is used in daily spoken communication, TV shows, songs and movies. In contrast to MSA, Arabic dialects are less closely related to Classical Arabic. DA is a mix of Classical Arabic and other ancient forms from different neighbouring countries that developed because of social interaction between people in Arab countries and people in the neighbouring countries (Biadsy et al., 2009).

There are many Arabic dialects that are spoken and written around the Arab world. The main Arabic dialects are: Gulf Dialect (GLF), Iraqi Dialect (IRQ), Levantine Dialect (LEV), Egyptian Dialect (EGY) and North African Dialect (NOR) as shown in Figure 1.

GLF is spoken in countries around the Arabian Gulf, and includes dialects of Saudi Arabia, Kuwait, Qatar, United Arab Emirates, Bahrain, Oman and Yemen. IRQ is spoken in Iraq, and it is a sub-dialect of GLF. LEV is spoken in countries around the Mediterranean east coast, and covers the dialects of Lebanon, Syria, Jordan, and Palestine. EGY includes the dialects of Egypt and Sudan. Finally, NOR includes the dialects of Morocco, Algeria, Tunisia and Libya (Alorifi, 2008; Biadsy et al., 2009; Habash, 2010).
For the time being, the researchers starting to work with Arabic dialect text, especially after the increasing use of Arabic dialect texts in informal settings such as social media as in the web, but almost available datasets for linguistics research are in MSA, especially in textual form (Zaidan and Callison-Burch, 2011). There is a lack of an Arabic dialects corpus, and no standardization in creating

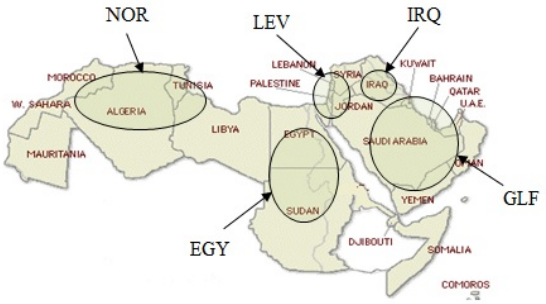

Figure 1: The Arab World.

an Arabic dialects corpus, so we tried to use Twitter and Facebook, the social applications that represent a dialectal text, because they attract a lot of people who freely write in their dialects. In addition, to cover the long dialect texts so we tried to use online commentary texts from the Arabic newspapers. The classification of dialects becomes an important pre-process step for other tasks, such as machine translation, dialect-to-dialect lexicons, and information retrieval (Malmasi et al., 2015). So, the next step after collecting data is annotate the text with the correct dialect tag to improve the accuracy of classifying Arabic dialect text.

In this paper, we present our methods to create a corpus of dialectal Arabic text by extracting tweets from Twitter based on coordinate points. Furthermore, we describe how to collect the comments from Facebook posts and online Arabic newspapers as a web source of a dialectal Arabic text. Then, we describe the new approach which used to annotate Arabic dialect texts. The paper is organized as follows: in section 2 we review related works on an Arabic dialects corpus, and annotation. Section 3 is divided into three subsections: in the first subsection, we present our method on how to extract tweets, the second subsection presents the methodology that we used to collect Facebook comments on timeline posts, the third subsection presents the approach was used to collect comments from online newspaper. Section 4 presents why annotation process is important, and describes the method used to annotate the collected dataset to build a corpus of Arabic dialect texts. Section 5 shows the total number of collected and annotated documents. Finally, the last section presents the conclusion and future work.

2. Related Work

Arabic dialect studies developed rapidly in recent months. However, any classification of dialects depends on a corpus to use in training and testing processes. There are many studies that have tried to create Arabic dialects corpora; however, many of these corpora do not cover the geographical variations in dialects. In addition, a lot of them are not accessible to the public. The following section describes the corpora that were built by the previous studies.

A multi dialect Arabic text corpus was built by (Almeman and Lee, 2013) using a web corpus as a resource. In this research, they focused only on distinct words and phrases which are common and specific to each dialect. They covered four main Arabic dialects: Gulf, Egyptian, North African and Levantine.
They collected 1,500 words and phrases by exploring the web and extracting each dialect's words and phrases, which must have been found in one dialect of the four main dialects. In the next step, they made a surveyed a native speaker for each dialect to distinguish between the words and confirm that words were used in that dialect only. After the survey, they created a corpus containing 1,000 words and phrases in the four dialects, including 430 words for Gulf, 200 words for North Africa, 274 words for Levantine and 139 words for Egyptian.

Mubarak and Darwish (2014) used Twitter to collect an Arabic multi-dialect corpus (Mubarak and Darwish, 2014). The researchers classified dialects as Saudi Arabian, Egyptian, Algerian, Iraqi, Lebanese and Syrian. They used a general query, which is lang:ar, and issued it against Twitter's API to get the tweets which were written in the Arabic language. They collected 175M Arabic tweets, then, extracted the user location from each tweet to classify it as a specific dialect according to the location.
Then, the tweets were classified as dialectal or not dialectal by using the dialectal words from the Arabic Online Commentary Dataset (AOCD) described in (Zaidan and Callison-Burch, 2014). Each dialectal tweet was mapped to a country according to the user location mentioned in the user's profile, with the help of the GeoNames geographical database (Mubarak and Darwish, 2014). The next step was normalization to delete any non-Arabic characters and to delete the repetition of characters. Finally, they asked native speakers from the countries identified as tweet locations to confirm whether each tweet used their dialects or not. At the end of this classification, the total tweets number about 6.5M in the following distribution: 3.99M from Saudi Arabia (SA), 880K from Egypt (EG), 707K from Kuwait (KW), 302K from United Arab Emirates (AE), 65k from Qatar (QA), and the remaining 8% from other countries such as Morocco and Sudan (Mubarak and Darwish, 2014).

Alshutayri and Atwell (2017) collected dialectal tweets from Twitter for country groups (5 groups) which are GLF, IRQ, LEV, EGY, and NOR, but instead of extracting all Arabic tweets as in (Mubarak and Darwish, 2014), the dialectal tweets were extracted by using a filter based on the seed words belonging to each dialect in the Twitter extractor program (Alshutayri and Atwell, 2017). The seed words are distinguished words that are used very common and frequently in one dialect and not used in any other dialects, such as the word (مصاري) (msary), which means "Money" and is used only in LEV dialect; we also used the word (دلوقتي)(dlwqty), which means "now" and is used only in EGY dialect, while in GLF speakers used the word (الحين) (Alhyn). In IRQ, speakers change Qaaf (ق) to (ك) so they say (وكت) (wkt), which means "time". Finally, for NOR, which is the dialect most affected by French colonialism and neighboring countries, speakers used the words (بزاف) (Bzaf) and (برشا) (brʃã), which mean "much". They extracted all tweets written in the Arabic language, and tracked 35 seed words all unigram in each dialect. In addition to the user location was used to show the geographical

location of the tweets, to be sure that tweets belong to this dialect. They collected 211K tweets with the total number of words equal to 3.6M words; these included 45K tweets from GLF, 40K from EGY, 45K from IRQ, 40K from LEV, and 41K from NOR.
Zaidan and Callison-Burch (2014) worked on Arabic Dialects Identification and focused on three Arabic dialects: Levantine, Gulf, and Egyptian. They created a large data set called the Arabic Online Commentary Dataset (AOCD) which contained dialectal Arabic content (Zaidan and Callison-Burch, 2014). Zaidan and Callison-Burch collected words in all dialects from readers' comments on the three on-line Arabic newspapers which are Al-Ghad from Jordan (to cover the Levantine dialect), Al-Riyadh from Saudi Arabia (to cover the Gulf dialect), and Al-Youm Al-Sabe from Egypt (to cover the Egyptian dialect). They used the newspapers to collect 1.4M comments from 86.1K articles. Finally, they extracted 52.1M words for all dialects. They obtained 1.24M words from Al-Ghad newspaper, 18.8M form Al-Riyadh newspaper, and 32.1M form Al-Youm Al-Sabe newspaper. In (Zaidan and Callison-Burch, 2014) the method of the annotation was used through the workers on Amazon's Mechanical Turk. They showed 10 sentences per screen. The worker was asked to label each sentence with two labels: the amount of dialect in the sentence, and the type of the dialect. They collected 330K labelled documents in about 4.5 months. But, compared to our method they pay to the workers a reward of $0.10 per screen. The total cost of annotation process was $2,773.20 in addition to $277.32 for Amazon's commission.

The last research used the text in Facebook to create corpus for sentiment analysis (Itani et al., 2017). The authors manually copying post texts which written in Arabic dialect to create news corpus collected from "Al Arabiya" Facebook page and arts corpus collected from "The Voice" Facebook page. Each corpus contained 1000 posts. They found that 5% of the posts could associated with a specific dialect while 95% are common to all dialect. After collecting the Facebook posts and comments in each post they started to preprocess the texts by removing time stamps and redundancy. In the last step, the texts were manually annotated by four native Arabic speakers' expert in MSA and Arabic dialects. The labels are: negative, positive, dual, spam, and neutral. To validate the result of the annotation step, the authors just accept the post which all annotators annotated it with same label. The total number of posts are 2000 divided into 454 negative posts, 469 positive posts, 312 dual posts, 390 spam posts, and 375 neutral posts.

3. The Arabic Dialects Corpora

In recent years, social media has spread between people as a result of the growth of wireless Internet networks and several social applications of Smartphones. These media sources of texts contain people's opinions written in their dialects which make it the most viable resources of dialectal Arabic. The following sections describe our method of collecting the Arabic dialect texts from Twitter, Facebook, and Online newspaper comments.

3.1. Twitter Corpus Creation

Twitter is a good resource to collect data compared to other social media because the data in Twitter is public, Twitter makes an API to help researchers to collect their data, and the ability to show other information, such as location (Meder et al., 2016). However, there is a lack of an available and reliable Twitter corpus which makes it necessary for researchers to create their own corpus (Saloot et al., 2016). Section 2 showed a method used to collect tweets based on seed terms (Alshutayri and Atwell, 2017) but, to cover all dialectal texts with different terms not just the seed terms, another method is used to collect tweets based on the coordinate points of each country using the following steps:

1. Use the same app that was used in (Alshutayri and Atwell, 2017) to connect with the Twitter API[1] and access the Twitter data programmatically.

2. Use the query lang:ar which extracts all tweets written in the Arabic language.

3. Filter tweets by tracking coordinate points to be sure that the Arabic tweets extracted from a specific area by specify the coordinate points (longitude and latitude) for each dialect area by using find latitude and longitude website (Zwiefelhofer, 2008). We specified the coordinate points for capital cities in North African countries, Gulf Arabian countries, Levantine countries, Egypt country, and Iraq country. In addition to the coordinates points of the famous and big cities in each country. The longitude and latitude coordinate points helped to collect tweets from the specified areas but to collect tweets with different subjects and contain several dialectal terms we ran the API at different time periods to cover lots of topics and events

4. Clean the tweets by excluding the duplicate tweets and deleting all emojis, non-Arabic character, all symbols such as (#,_,"), question mark, exclamation mark, and links, then label each tweet with its dialect based on the coordinate points which used to collect this tweet.

Using this method to collect tweets based on coordinate points for one month, obtained 112K tweets from different countries in the Arab world. The total number of tweets after the cleaning step and deleting the redundant tweets equal to 107K tweets, divided between dialect as in table 1. Figure 2 shows the distribution of tweets per dialect. We noticed that we can extract lots of tweets from the GLF dialect in comparison to LEV, IRQ, NOR and EGY and this is because Twitter is not popular in these dialects' countries as Facebook in addition to the internal disputes in some countries which have affected the ease of use of the Internet.

3.2. Facebook comments Corpus Creation

Another source of Arabic dialect texts is Facebook which consider as one of the famous social media applications in

[1] http://apps.twitter.com

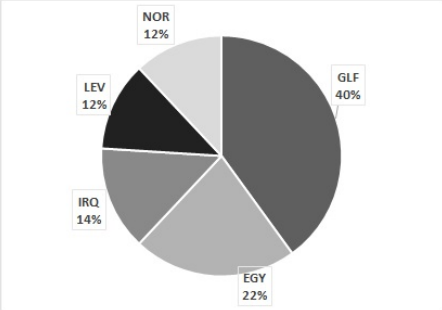

Figure 2: The distribution of dialectal tweets based on location Points

the Arab world, and lots of users writing in Facebook using their dialects. We collected comments by following the steps below:

1. At the beginning to collect the Facebook comments, the Facebook pages which used to scrape timeline posts and its comments are chosen by using Google to search about the most popular Arabic pages on Facebook in different domains such as, sport pages, comedy pages, channel and program pages, and news pages.
2. The result from first step which was a list of Arabic pages are explored and checked for every page to see if it contains lots of followers, posts and, comments, then created a final list of pages to scrape posts.
3. Create an app which connects with the Facebook Graph API[2] to access and explore the Facebook data programmatically. The app worked into steps:
   (a) First, collected all posts of the page started from the page establish date until the day that the app was executed. The result of this step is a list of posts id for each page which help to scrape comments from each post in addition to some metadata for each post may help other research, for example, post type, post link, post published date, and the number of comments in each post.
   (b) Then, the results of the previous step for each page are used to scrape comments for each post based on the post id. The result of this step is a list of comment messages and some metadata such as, comment id, post id, parent id of the comment if the comment is a replayed to another comment, comment author name and id, comment location if the author add the location information in his/her page, comment published date, and the number of likes for each comment.
4. In the third step, the comment's id and message which extracted from the previous step is labeled with the dialect based on the country of the Facebook page which used to collect the posts from it.
5. Finally, clean the comment messages by deleting the duplicate comments, and delete all emojis, non-Arabic character, all symbols such as (#,_,"), question mark, exclamation mark, and links.

[2]https://developers.facebook.com/

The API to scrap Facebook was ran for one month and at the end of this experiment, we obtained a suitable quantity of text to create Arabic dialect corpus and use it in classification process. The total number of collected posts equal to 422K and the total number of collected comments equal to 2.8M. After the cleaning step we got 1.3M comments, divided into dialects as in table 1.
We tried to make our corpus balanced by collecting the same number of comments for each dialect, but the problem that we did not find Facebook pages rich with comment for some country such as Kuwait, UAE, Qatar, and Bahrain. Figure 3 is a chart shows the percentage of the number of comments collected for each dialect, and we noticed that the number of comments in IRQ and GLF are less compared with other dialect due to the fewest number of Facebook pages were found to cover these dialects. In addition, unpopularity of Facebook application in Gulf area in comparison with Twitter application, and the bad internet coverage in Iraq country due to impact of war in Iraq. While, we collected a good number of comments for NOR dialect as some in North Africa countries Facebook is more popular than Twitter.

Dialect	No. of Tweets	No. of Facebook comments
GLF	43,252	106,590
IRQ	14,511	97,672
LEV	12,944	132,093
NOR	13,039	212,712
EGY	23,483	263,596

Table 1: The number of tweets and Facebook comments in each dialect

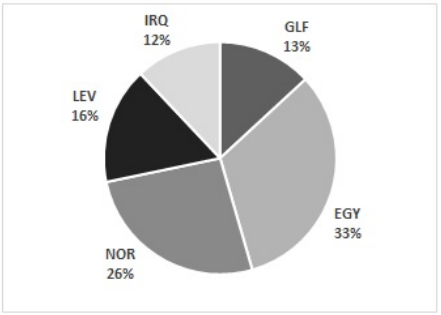

Figure 3: The percentage of the number of Facebook comments collected for each dialect.

3.3. Online Newspaper Comments Corpus Creation

The readers' comments on online newspaper are another source of dialectal Arabic text. An online commentary is chosen as a resource to collect data because it is public,

structured and formatted in a consistent format which make it easy to extract (Zaidan and Callison-Burch, 2011). Furthermore, we can automatically collect large amounts of data updated every day with new topics. The written readers' comments were collected from 25 different Arabic online newspaper based on the country which issued each of the newspapers for example, Ammon for Jordanian comments (LEV dialect), Hespress for Moroccan comments (NOR dialect), Alyoum Alsabe' for Egyptian comments (EGY dialect), Almasalah for Iraqi comments (IRQ dialect), and Ajel for Saudi comments (GLF dialect). This step was done by exploring the web to search about a famous Online newspaper in the Arab countries in addition to asking some native speakers about the common newspaper in their country.

We tried to make our data set balanced by collecting around 1000 comments for each dialect. Then, classify texts and label it according to the country that issue the newspaper. In addition, to ensure that each comment belongs to the dialect which was labelled to it, the comments are automatically revised by using the list of seed words which created to collect tweets by checking each word in the comment and decide to which dialect it belongs. However, we found some difficulty with comments because lots of comments, especially from GLF dialect are written in MSA, which affects the results of automatic labelling so we found that we also need to re-label the comments manually using an annotation tools. The last step was cleaning the collected comments by removing the repeated comments and any unwanted symbols or spaces.

Around 10K comments are collected by crawling the newspaper sites during a two-month period. The total number of words equal to 309,994 words; these included 90,366 words from GLF, 31,374 from EGY, 43,468 from IRQ, 58,516 from LEV, and 86,270 from NOR. Figure 4 shows the distribution of words per dialect. We planned to collect readers' comments from each country in the five groups of dialects. For example, comments from Saudi Arabia newspaper and comments from Kuwait newspaper to cover the Gulf dialect and so on for all dialects, but the problem that in some countries such as Lebanon and Qatar we did not find lots of comments.

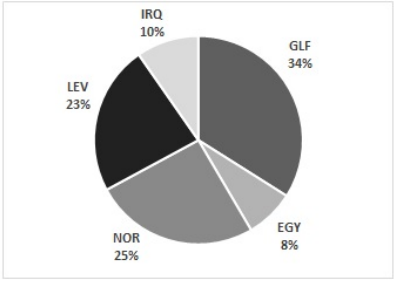

Figure 4: The distribution of words per dialect collected from Newspaper.

4. The Annotation Process

4.1. Importance of the Annotation Process

We participated in the COLING 2016 Discriminating Similar Languages (DSL) 2016 shared task (Alshutayri et al., 2016), where Arabic dialect text used for training and testing were developed using the QCRI Automatic Speech Recognition (ASR) QATS system to label each document with a dialect (Khurana and Ali, 2016) (Ali et al., 2016). Some evidently mislabelled documents were found which affected the accuracy of classification; so, to avoid this problem a new text corpus and labelling method were created.

In the first step of labelling the corpus, we initially assumed that each tweet could be labelled based on the location appears in the user's profile and the location points which used to collect the tweets from Twitter. As for the comments were collected from online newspapers, each comment labelled based on the country where the newspaper is published. Finally, for the comments collected from Facebook posts, each comment labelled based on the country of the Facebook page depended on the nationality of the owner of the Facebook page if it is a famous public group or person. However, through the inspection of the corpus, we noticed some mislabelled documents, due to disagreement between the locations of the users and their dialects, and the nationality of the page owner and the comments text. So, must be verify that each document is labelled with the correct dialect.

4.2. Method

To annotate each sentence with the correct dialect, 100K documents were randomly selected from the corpus (tweets and comments), then created an annotation tool and hosted this tool in a website.
In the developed annotation tool, the player annotates 15 documents (tweets and comments) per screen. Each of these documents is labelled with four labels, so the player must read the document and make four judgments about this document. The first judgment is the level of dialectal content in the document. The second judgment is the type of dialect if the document not MSA. The third judgment is the reason which makes the player to select this dialect. Finally, the fourth judgment if the reason selected in the third judgment is dialectal terms; then in the fourth judgment the player needs to write the dialectal words were found in the document.
The following list shows the options under each judgment to let the player choose one of them.

- The level of dialectal content
  - MSA (for document written in MSA)
  - Little bit of dialect (for document written in MSA but it contains some words of dialect less than 40% of text is dialect, see figure 5)
  - Mix of MSA and dialect (for document written in MSA and dialect around 50% of text is MSA (code-switching)), see figure 6
  - Dialect (for document written in dialect)

- The type of dialect if the document written in dialect
  - Egyptian
  - Gulf
  - Iraqi
  - Levantine
  - North African
  - Not Sure
- The reason that make this document dialectal
  - Sentence Structure
  - Dialectal Terms
- The words which identify the dialect (we need to use these word as a dictionary for each dialect)

To annotate the collected data, an interface was built as a web page `http://www.alshutayri.com/index.jsp` to display a group of Arabic documents randomly selected from the dataset.

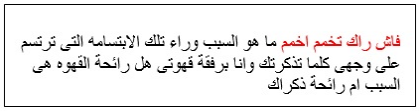

Figure 5: Example of document labeled as littel bit of dialect.

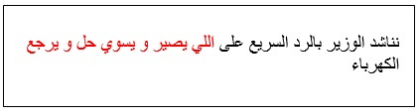

Figure 6: Example of document labeled as mix of MSA and dialect.

Each page displays 15 documents randomly selected from the dataset. The first label indicates the amount of dialectal content in the document to decide whether the document is MSA or contains dialectal content. If the document is MSA the other labels will be inactive, and the player needs to move to the next document. But, if the document is not MSA, then all labels are required. The second label specifies the document dialect if it is one of the five dialects (EGY, GLF, LEV, IRQ, and NOR), or Not Sure if the document written using dialect but difficult to decide which dialect. The third and fourth labels to explain the causes to choose the selected dialect: for example, the sentence structure if the words in the document are all MSA words, but the structure of the sentence is not based on the MSA grammar rules, and/or the dialectal terms which are famous words help to identify the dialect. In fact, there is no agreed standard for writing Arabic dialects because MSA is the formal standard form of written Arabic (Elfardy and Diab, 2012); therefore, some documents apparently contain only MSA vocabulary but are annotated as dialect based on non-standard sentence structure.

Before submitting the annotated documents, the mother dialect must be chosen. This may help to decide which annotated document must be accepted if one document has different annotations. Finally, by submitting the annotated documents the score will be shown in the screen by comparing the labelled documents with our pre-labelled sample.

As a control to be sure that the player reads the document before selecting the options, three MSA documents collected from a newspaper articles (Al-Sulaiti and Atwell, 2004), were mixed with 12 documents selected from the dataset; so, these three MSA documents used as a control because they must be labelled as MSA, so if the player labels all the three MSA documents as a dialect then the player's submitted documents are not counted in the annotated corpus. Furthermore, to verify the annotation process, each document is redundantly being annotated three times.

5. Result

The corpus covers five Arabic dialects: GLF, EGY, NOR, LEV, and IRQ. It consists of tweets from Twitter, Comments from Online Newspaper, and comments from Facebook. The tweets were collected using two methods: based on seed terms as we presented in (Alshutayri and Atwell, 2017), and based on coordinate points. The comments from Facebook were collected based on the country of the Facebook page as well as comments from Newspaper based on the country that issued the newspaper. After the collection step, the texts from the three different sources are revised and processed based on the following criteria:

- Exclude any documents if the writer of tweet or comment write his nationality which conflict with the label of the document based on the method which used to collect this document, see figure 7.
- Exclude any duplicated documents which are appear frequently, especially in tweets due to retweeting or copying.
- Keep the length for each document as written.

Figure 7: Example of the excluding documents from the corpus.

The final version of the corpus after applying the previous criteria, contains 1.1M documents; they include 812K Facebook comments, 9K online newspaper comments, and 266K Twitter tweets; 180K based on seed terms, and 86K

based on coordinate points. According to these numbers, we found that Facebook gives lots of comments in comparison to Twitter and Online newspaper, because using Facebook to scrape all posts for a specific Facebook page got all posts from the beginning of the page creation, so for each post lots of comments are collected from different users with a good amount of different words. While on Twitter it is difficult to recognize a specific account to collect all that account's tweets because we want to cover many users with different tweets topics and dialects. So, the program worked randomly at every day for a specific period ranging from 4-6 hours to collect all tweets written at this time. Figure 8 shows the distribution of dialectal content in the annotated documents. Table 2 presents the number of types in each dialect from all sources.

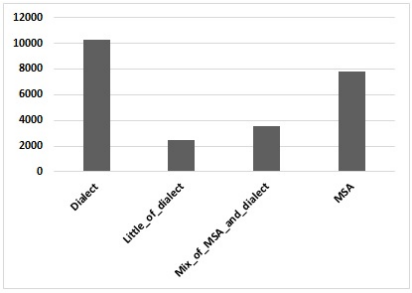

Figure 8: The result of the level of dialectal content in the annotated documents.

Source	GLF	EGY	NOR	LEV	IRQ
Tweets Based on Seed Terms	51,527	40,956	43,555	62,463	56,429
Tweets Based on Coordinate Points	77,302	48,230	96,901	38,705	35,901
Facebook Comments	153,146	211,891	346,298	175,216	131,542
Comments from Newspaper	28,949	12,654	27,585	20,869	14,907

Table 2: The number of types in each dialect in different sources

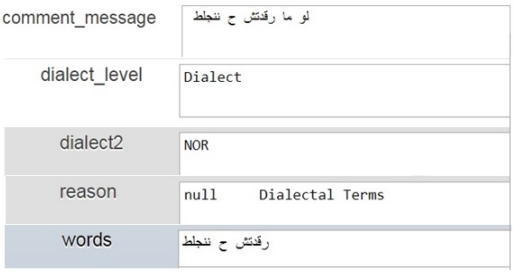

Figure 9: Result of the Annotatted Document.

Figure 9 shows the result of one annotated document in the corpus. Each document is labelled with four labels: the first label is the dialect level, which is an option from three choices: little_of_dialect, Mix_of_MSA_and_dialect, or Dialect. The second label is the specific dialect which is one of the five dialects: GLF, EGY, LEV, IRQ, or NOR. The third label shows the reasons that help to identify the document's dialect. The last label shows the dialectal words which help to identify the document's dialect. The document in figure 9 annotated as NOR dialect based on some dialectal terms were written in the words cell.

We launched the website via Twitter and WhatsApp at the beginning of August 2017. At the time of paper submission, we have been running the annotation website for around four months, and we have accumulated 24K annotated documents with total numbers of words equal to 587K. The number of users (players) equal to 1,575 from different countries around the world. To measure the quality of the annotation, the inter-annotator agreement was calculated using Fleiss Kappa (Fleiss, 1971) to calculate the annotator agreement for more than two annotators. The result equal to 0.787 around 79% which is substantial agreement according to (Landis and Koch, 1977). For our immediate research on Arabic dialects classification the annotated documents which we have already collected could be sufficient, but we decided to continue with this experiment to collect a larger annotated Arabic dialect text corpus.

6. Conclusion

This paper has explored social media text as a reference for Arabic dialects. We divided the Arab countries into five groups, one for each of the five main dialects: Gulf, Iraqi, Egyptian, Levantine, and North African. The text was classified based on the seed words that are spoken in one dialect and not in the other dialects. In addition to the user location which help to enhance dialect classification and specify the country and dialect to which each tweet belongs. In addition, we scraped Facebook posts and extracted all comments from these posts based on the famous Facebook pages in the Arab world countries. The extracted comments classified based on the nationality of the Facebook owner. Furthermore, online comments in Newspaper considers as a good source of dialectal Arabic, especially if the article talking about things that are interesting to the community of this country, for example living conditions and a high cost of living, art, or sport because if the topic of the article is about political news lots of readers comment using MSA instead of their dialect, so lots of comments mix of MSA and dialect. The comments were classified based on the country that issued the newspaper.

In general, the social media can be used as a reference to collect an Arabic dialects corpus, but to make our corpus balanced we tried to run the extractor in one dialect more than another as we noticed that Twitter is more popular in Arabian Gulf area which help us to collect lots of tweets for GLF dialect whereas the fewer tweets from North Africa countries and Iraq. In comparison with Twitter, Facebook is more popular in North African.

In this paper, we presented a new approach to annotate the dataset were collected from Twitter, Facebook, and Online Newspaper for the five main Arabic dialects: Gulf, Iraqi, Egyptian, Levantine and North African. The annotation

website was created as an online game to gather more users who talk different Arabic dialects and free to pay in comparing with other crowdsourcing websites. This experiment is a new approach help to annotate a sufficient dataset for text researches in Arabic dialect classification. The number of users has decreased now in comparison with the beginning because we need to redistribute the website widely.

In the future work we will explore another source of the Arabic dialect text such as WhatsApp application, or YouTube comments to cover most of sources and build a corpus including different sources of the texts. In addition, we could modify the interface to be more attractive and easy to explore. In addition, we could make this annotation game as an application can be downloaded in the smart phones and tablets.

7. Bibliographical References

Al-Sulaiti, L. and Atwell, E. (2004). Designing and developing a corpus of contemporary arabic.

Ali, A., Dehak, N., Cardinal, P., Khurana, S., Yella, S. H., Glass, J., and, P. B., and Renals, S. (2016). Automatic dialect detection in arabic broadcast speech. *Interspeech2016*, pages 2934–2938.

Almeman, K. and Lee, M. (2013). Automatic building of arabic multi dialect text corpora by bootstrapping dialect words.

Alorifi, F. S. (2008). *Automatic Identification of Arabic Dialects Using Hidden Markov Models*. Thesis.

Alshutayri, A. and Atwell, E. (2017). Exploring twitter as a source of an arabic dialect corpus. *International Journal of Computational Linguistics (IJCL)*, 8(2):37–44.

Alshutayri, A., Atwell, E., Alosaimy, A., Dickins, J., Ingleby, M., and Watson, J. (2016). Arabic language weka-based dialect classifier for arabic automatic speech recognition transcripts. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 204–211.

Biadsy, F., Hirschberg, J., and Habash, N. (2009). Spoken arabic dialect identification using phonotactic modeling, 31 March.

Elfardy, H. and Diab, M. (2012). Token level identification of linguistic code switching. In *Proceedings of COLING*, pages 287–296.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Habash, N. Y. (2010). *Introduction to Arabic Natural Language Processing*. Morgan and Claypool.

Itani, M., Roast, C., and Al-Khayatt, S. (2017). Corpora for sentiment analysis of arabic text in social media.

Khurana, S. and Ali, A. M. (2016). Qcri advanced transcription system (qats) for the arabic multi-dialect broadcast media recognition: Mgb-2 challenge. *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 292–298.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Malmasi, S., Refaee, E., and Dras, M. (2015). Arabic dialect identification using a parallel multidialectal corpus. *Pacific Association for Computational Linguistics*, pages 203–211.

Meder, T., Nguyen, D., and Gravel, R. (2016). The apocalypse on twitter. *Digital Scholarship in the Humanities*, 31(2):398–410.

Mubarak, H. and Darwish, K. (2014). Using twitter to collect a multi-dialectal corpus of arabic, October 25.

Saloot, M. A., Idris, N., Aw, A., and Thorleuchter, D. (2016). Twitter corpus creation: The case of a malay chat-style-text corpus (mcc). *Digital Scholarship in the Humanities, Vol. 31, No. 2,*, 31(2):227–243.

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*, pages 37–41. Association for Computational Linguistics.

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

Zwiefelhofer, D. B. (2008). Find latitude and longitude, 2017.

Diacritization of Moroccan and Tunisian Arabic Dialects: A CRF Approach

Kareem Darwish*, Ahmed Abdelali*, Hamdy Mubarak*, Younes Samih†, Mohammed Attia⋆
*QCRI, †University of Dusseldorf, ⋆Google Inc.
{kdarwish, aabdelali, hmubarak}@qf.edu.qa, samih@phil.hhu.de, attia@google.com

Abstract

Arabic is written as a sequence of consonants and long vowels, with short vowels normally omitted. Diacritization attempts to recover short vowels and is an essential step for Text-to-Speech (TTS) systems. Though Automatic diacritization of Modern Standard Arabic (MSA) has received significant attention, limited research has been conducted on dialectal Arabic (DA) diacritization. Phonemic patterns of DA vary greatly from MSA and even from one another, which accounts for the noted difficulty of mutual intelligibility between dialects. In this paper we present our research and benchmark results on the automatic diacritization of two Maghrebi sub-dialects, namely Tunisian and Moroccan, using Conditional Random Fields (CRF). Aside from using character n-grams as features, we also employ character-level Brown clusters, which are hierarchical clusters of characters based on the contexts in which they appear. We achieved word-level diacritization errors of 2.9% and 3.8% for Moroccan and Tunisian respectively. We also show that effective diacritization can be performed out-of-context for both sub-dialects.

Keywords: Arabic, Dialects, Vowelization, Diacritization

1. Introduction

Different varieties of Arabic are typically written without diacritics (short vowels). Arabic readers disambiguate words in context and mentally restore diacritics to pronounce words correctly. For Modern Standard Arabic (MSA), diacritics serve dual functions. While word-internal diacritics are phonemic in nature and dictate correct pronunciation and lexical choice, final vowels on words (a.k.a case endings) indicate syntactic role. However, dialects overwhelming use *sukun* as a neutral case-ending for all words, eliminating the need for disambiguating syntactic roles. Thus, dialectal diacritic recovery involves restoring internal-word diacritics only. The task of diacritic restoration is crucial for applications such as text-to-speech (TTS) to enable the proper pronunciation of words.

In this paper, we present new state-of-the-art Arabic diacritization of two sub-dialects of Maghrebi, namely Moroccan and Tunisian, using Conditional Random Fields (CRF) sequence labeling. We trained our CRF sequence labeler using character n-grams as well as character-level Brown clusters. In our context, Brown clusters would bin together characters that appear in similar contexts, which would improve the generalization of the training set. We explore mono-dialectal training as well as cross-dialectal and joint training. Using mono-dialectal training, we achieve word error rates of 2.9% for Moroccan and 3.8% for Tunisian. Though both sub-dialects are orthographically similar, we show that cross-dialectal and joint training lead to significant increases in diacritization errors due to the phonetic divergence of the sub-dialects. Thus, dialectal TTS needs to be tuned for specific sub-dialects.

The contributions of this work are:

• To the best of our knowledge, this is the first work on the diacritization of Maghrebi Arabic, which helps shed more light on the properties of some spoken variants of Arabic and providing benchmark results.

• We show that diacritization can be performed with high accuracy for words out of context.

• We explore the use of cross dialect and joint dialect training between Moroccan and Tunisian, highlighting the orthographic and phonetic similarities and dissimilarities of both sub-dialects.

2. Background

Significant research has addressed diacritic restoration/recovery or diacritization for Arabic, mostly MSA, and some other Semitic languages which are typically written without short vowels. Diacritization is essential for a variety of applications such as TTS and language learning. MSA diacritization involves internal-word diacritization to disambiguate meaning and case ending recovery based on syntactic role. Recovering the case ending is typically significantly harder than core word diacritization. Dialects have mostly eliminated case endings, using the silence diacritic *sukun* instead. Many approaches have been used for internal-word diacritization of MSA such as Hidden Markov Models (Gal, 2002; Darwish et al., 2017), finite state transducers (Nelken and Shieber, 2005), character-based maximum entropy based classification (Zitouni et al., 2006), and deep learning (Abandah et al., 2015; Belinkov and Glass, 2015; Rashwan et al., 2015). Darwish et al. (2017) compared their system to others on common test set. They achieved a word error rate of 3.29% compared 3.04% for Rashwan et al. (2015), 6.73% for Habash and Rambow (2007), and 14.87 for Belinkov and Glass (2015). Azmi and Almajed (2015) survey much of the literature on MSA diacritization.

Concerning dialectal diacritization, the literature is rather scant. Habash et al. (2012) developed a morphological analyzer for dialectal Egyptian, which uses a finite state transducer that encodes manually crafted rules. They report an overall analysis accuracy of 92.1% without reporting diacritization accuracy specifically. Khalifa et al. (2017) developed a morphological analyzer for dialectal Gulf verbs, which also attempts to recover diacritics. Again, they did not specifically report diacritization accuracy. Jarrar et al. (2017) annotated a corpus of dialectal Palestinian contain-

ing 43k words. Annotation included text diacritization. In the aforementioned papers, the authors used CODA, a standardized dialectal spelling convention. Other recent work on dialects attempted to perform different processing, such as segmentation, without performing any spelling standardization (Eldesouki et al., 2017; Samih et al., 2017). Diacritization without standardizing spelling is highly advantageous, and thus we pursue character level models in this paper.

3. Data

We were able to obtain two translations of the New Testament into two Maghrebi sub-dialects, namely Moroccan[1] and Tunisian[2] dialects. Both of them are fully diacritized and contain 8,200 verses each. Table 1 shows the data size, and Table 2 gives a sample verse from both dialects, MSA, and the English translation. We split the data for 5-fold cross validation, where training splits were further split 70/10 for training/validation. Given the training portions of each split, Figure 1 shows the distribution of the number of observed diacritized forms per word. As shown, 89% and 82% of words have one diacritized form for Moroccan and Tunisian respectively. We further analyzed the words with more than one form. The percentage of words where one form was used more than 99% of time was 53.8% and 55.5% for Moroccan and Tunisia respectively. We looked at alternative diacritized forms for this group and we found that the less common alternatives are cases were default diacritics (ex. *fatha* before *alef* – روما (rwmA) vs. رُومَا (ruwmaA) – meaning "Rome")[3] are dropped while they are generally present. Similarly, the percentage of words where the most frequent form was used less than 70% was 6.1% and 8.5% for Moroccan and Tunisian respectively. Aside from the cases where a surface form can have multiple possible diacritics (ex. الْحُكَامْ (AloHokaAmo – "the judging") – vs. الْحُكَّامْ (AloHuk∼aAmo – "the rulers")), we found frequent cases where a dicritized form has a *shadda–sukun* combination and another has just *sukun* (ex. يْحَزَّجُو (yoxar∼ojuw) vs. يْخَزْجُو (yoxarojuw) – "to drive out") and others were different diacritized forms would have nearly identical pronunciation (ex. يْرِيبْ (yoriybo) vs. يْرَيَّبْ (yoray∼obo) – "to destroy"). Further, we used the most frequent diacritized form for each word, and we automatically diacritized the training set. Doing so, the word error rate on the training set was 0.9% and 1.1% for Moroccan and Tunisian respectively. This indicates that diacritizing words out of context can achieve up to 99% accuracy (1% word error rate). We compared this to the MSA version of the same Bible verses (132,813 words) and a subset of diacrtized MSA news articles of comparable size (143,842 words) after removing case-endings. As Table 3 shows, MSA words, particularly for the Bible, have many more possible diacritized forms, and picking the most frequent diacritized form leads to significantly higher word error rate compared to dialects.

[1]Translated by Morocco Bible Society
[2]Translated by United Bible Societies, UK
[3]We use Buckwalter encoding to transliterate Arabic words.

Dialects	No. of Words
Moroccan	134,324
Tunisian	131,923

Table 1: Dialectal data size

Lang.	Verse (Colossians 3:20)
Moroccan	آ الْوْلَادْ، طِيعُو وَالِدِيكُمْ فْكُلْشِي
Tunisian	يَا الوْلَادْ، طِيعُوا وَالْدِيكُمْ فِي كُلْ شَيْء
MSA	أَيُّهَا ٱلْأَوْلَادُ، أَطِيعُوا وَالِدِيكُمْ فِي ٱلرَّبِّ
English	Children, obey your parents in all things

Table 2: Sample verse from diacritized dialectal Bibles

We compared the overlap between training and test splits. Figure 2 shows that a little over 93% of the test words were observed during training. If we use the most frequent diacritized forms observed in training, we can diacritize 92.8% and 92.0% of Moroccan and Tunisian words respectively. Thus, the job of a diacritizer is primarily to diacritize words previously unseen words, rather than to disambiguate between different forms. We also compared the cross coverage between the Moroccan and Tunisian datasets. As Figure 3 shows, the overlap is approximately 61%, and the diacritized form in one dialect matches that of the other dialect less than two thirds of the time. This suggests that cross dialect training will yield suboptimal results.

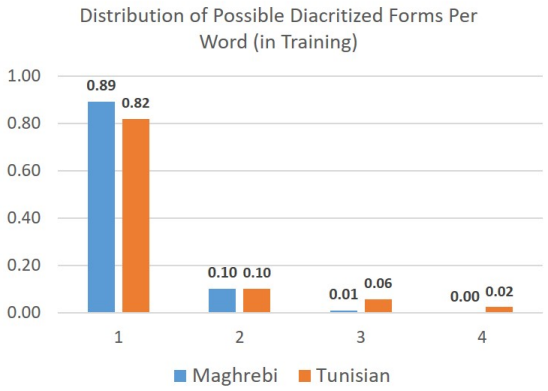

Figure 1: Distribution of the number of diacritized forms per word in training parts

There are 14 diacritics in MSA that Arabic letters can carry[4] in addition to *EMPTY* diacritic which is used for long vowels and sometimes for cases like the definite determiner ال (meaning "the"). In Moroccan, an extra diacritic is also used, namely *shaddah–sukun*. The distributions of different diacritics in Moroccan, Tunisian, and the corresponding MSA of the Bible data are shown in Figure

[4]https://en.wikipedia.org/wiki/Arabic_script_in_Unicode

	Bible	News
Most Freq	92.1	92.8
No. of Seen Forms		
1	51.7	69.0
2	20.4	26.8
3	13.5	2.9
4	7.1	1.1
≥5	7.3	0.1

Table 3: Distribution of the number of dicaritized forms per word for MSA

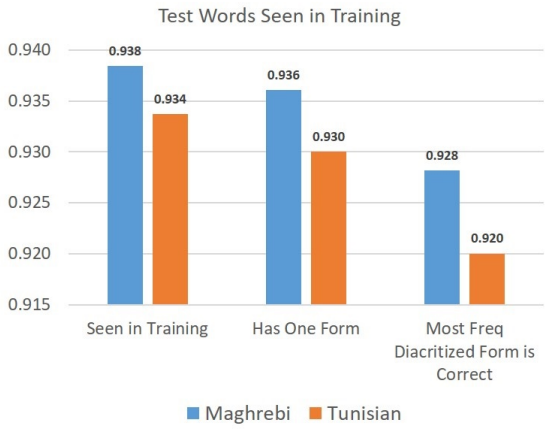

Figure 2: Overlap between train and test parts.

4. Generally, both Moroccan and Tunisian have comparable distributions, and they are different than of MSA. Also, while 34% and 26% of the letters have *sukun* in Moroccan and Tunisian respectively, only 4% of letters in MSA have *sukun*.

Figures 5 and 6 show distributions of diacritics for first letter (prompt) and last letter (typically case ending indicating grammatical function) in words to show how diacritization of Moroccan and Tunisian differ from MSA. In MSA, words cannot start with a letter with *sukun* because there is no morphological templatic pattern that starts with *sukun*. However, in Moroccan and Tunisian, 43% and 23% of the words start with *sukun*. For the last diacritic, MSA case endings can take many values. Conversely, Moroccan and Tunisian case endings are overwhelmingly either *sukun* (57% and 53%) or *EMPTY* (37% and 45%) respectively.

It is worth mentioning that in our corpus, the maximum number of *sukun* in a word is 6 for both dialects with words like وْلْبْلَايْض (wololobolaAyoSo – "and places") and وْنَبْعْثُوهُمْلْكُمْ (wonaboEovuhumolokumo – "and we send them to you") compared to only a maximum of 3 *sukun* in MSA words like فَأَشْفِيهِمْ (fa>u$ofiyhimo – "I will heal them"). Also, 23% of Moroccan words and 7% of Tunisian words have consecutive *sukun*, and the maximum number of consecutive *sukun* in Moroccan is 5, as in وْفْتْلْتْ (wofotoloto – meaning "and in three"), compared to only 2 for Tunisian, as in تِتْضْرَبْشْ (titoDorabo$o – "will not hit"). In the MSA Bible, there is only one word that has 2 consecutive sukun, namely سِمِيْرْنا (simiyorokA – a foreign named-entity "Smyrna"), because no words of Arabic origin are allowed to have 2 consecutive *sukun*. If two *sukun* happen to appear consecutively, MSA diacritization rules convert the first *sukun* to either *fatha* or *kasra*.

The Word ومشى (wm$Y – "and he walked") is an example of words that are written the same in Moroccan, Tunisian, and MSA with the same meaning but with different diacritization and hence pronunciation: وْمْشَى (womo$aY) in Moroccan; وِمْشَى (wimo$aY) in Tunisian; and وَمَشَى (wama$aY) in MSA. All the above indicate that using an MSA diacritizer to diacritize Moroccan or Tunisian would lead to high word error rate, because they follow different diacritization patterns and rules.

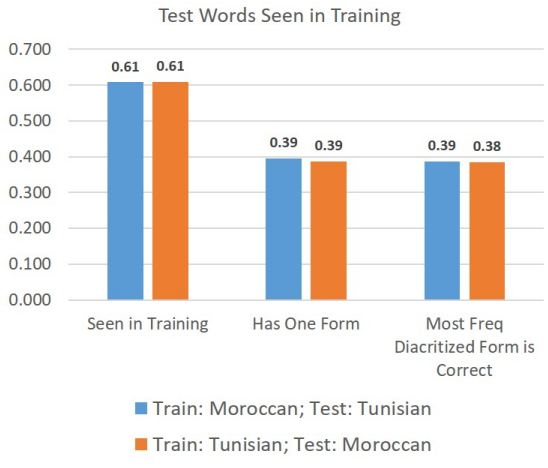

Figure 3: Overlap between train and test parts.

4. Proposed Approach: Linear Chain CRF

The effectiveness of CRFs (Lafferty et al., 2001) has been shown for many sequence labeling tasks, such as POS tagging and named entity recognition. CRFs effectively combine state-level features with transition features. They are simple and well-understood, and they usually provide efficient models with close to state-of-the-art results. Thus, CRF is a potentially effective method to apply to this task. For all the experiments, we used the CRF++ implementation of a CRF sequence labeler with L2 regularization and default value of 10 for the generalization parameter "C".[5] In our setup, our goal is to tag each character of every word with the appropriate diacritic, where character-level diacritics are our labels. For features, given a word of character sequence c_n ... c_{-2}, c_{-1}, c_0, c_1, c_2 ... c_m, we used a combination of character n-gram features, namely unigram (c_0), bigrams (c^0_{-1}; c^1_0), trigrams (c^0_{-2}; c^1_{-1}; c^2_0), and 4-grams (c^0_{-3}; c^1_{-2}; c^2_{-1}; c^3_0).

[5] https://github.com/taku910/crfpp

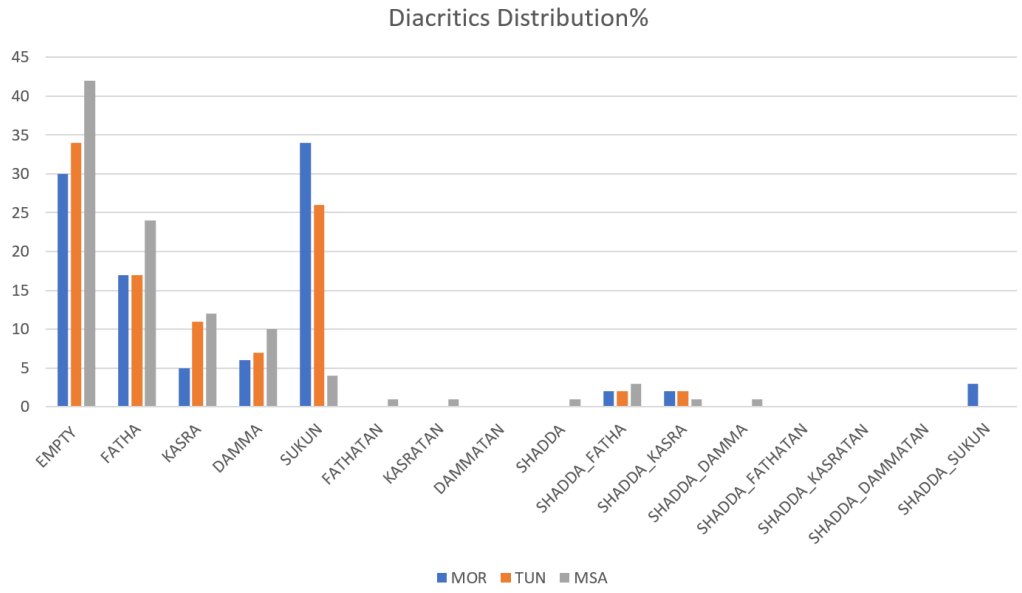

Figure 4: Diacritics Distribution in Moroccan, Tunisian, and MSA

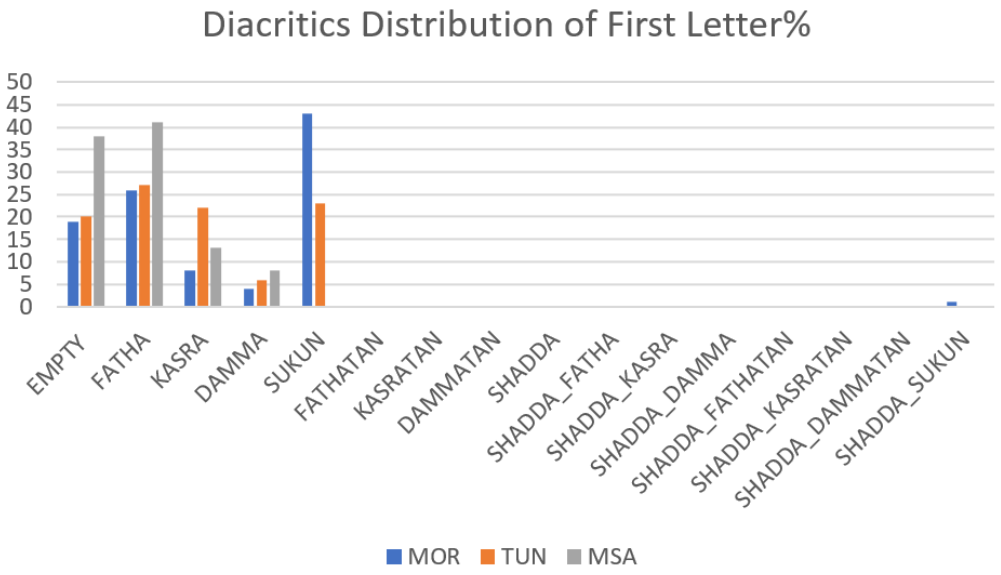

Figure 5: Diacritics Distribution of First Letter

Another feature that may potentially help our sequence labeling to generalize is the use of character level Brown clusters (Brown et al., 1992), which are hierarchical clusters of tokens based on the contexts in which they appear. They have been shown to improve many NLP tasks such as POS tagging (Owoputi et al., 2013). The rationale for using it here is that some characters may appear in similar contexts and would hence have similar diacrtics. The advantage is that Brown clusters can be learned from unlabeled texts. We generated 25 character clusters from the training part for each fold using the implementation of Liang (2005). When using Brown clusters, we used the aforementioned character n-gram features in addition to an identical set of features where we replace characters with their corresponding Brown cluster tags. Given that the vast majority of dialectal words have only one possible diacritized form, the CRF is trained on individual words out of context.

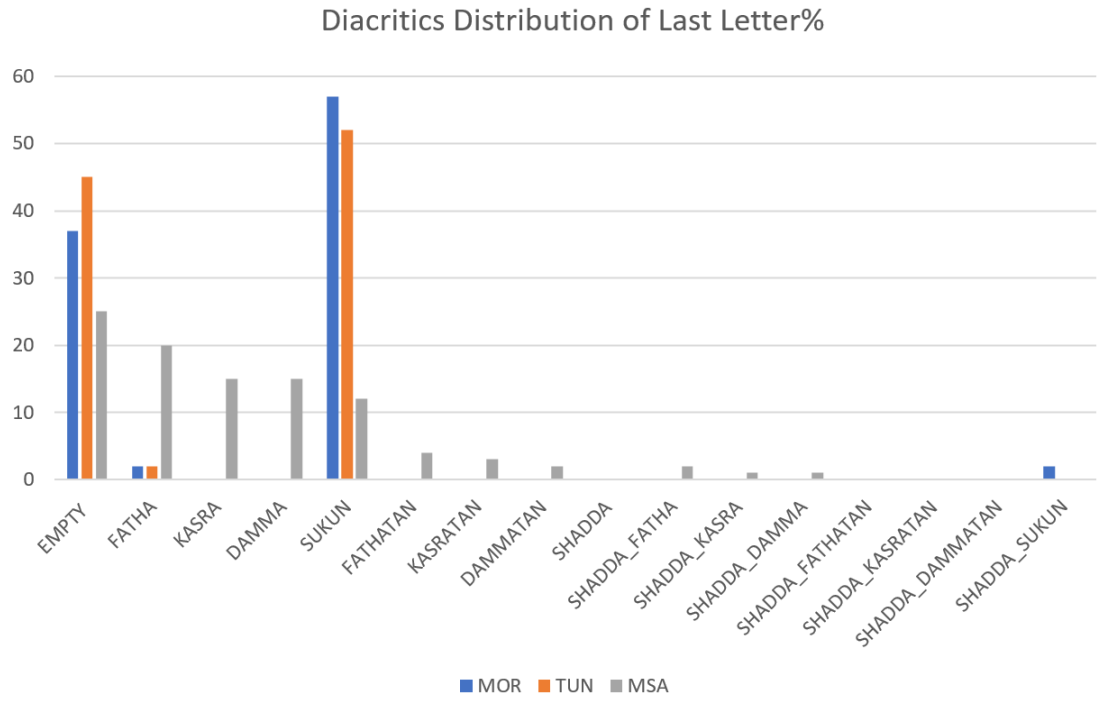

Figure 6: Diacritics Distribution of Last Letter

Training Set	Test Set	Error Rate Character	Word
(a) Uni-dialectal Training			
Moroccan	Moroccan	1.1	3.1
Tunisian	Tunisian	1.8	4.0
(b) Cross Training			
Moroccan	Tunisian	17.2	**43.3**
Tunisian	Moroccan	17.9	**43.9**
(c) Combined Training			
Combined	Moroccan	3.0	8.7
Combined	Tunisian	4.8	13.8

Table 4: CRF character n-grams results – Average across all folds

Training Set	Test Set	Error Rate Character	Word
(a) Uni-dialectal Training			
Moroccan	Moroccan	1.1	2.9
Tunisian	Tunisian	1.7	3.8
(b) Cross Training			
Moroccan	Tunisian	20.1	47.0
Tunisian	Moroccan	20.8	48.9
(c) Combined Training			
Combined	Moroccan	12.6	34.2
Combined	Tunisian	9.5	23.8

Table 5: CRF Results with Brown clusters – Average across all folds

5. Results

As shown in Figure 2, our baseline uses the most frequently seen diacritized form that is observed in training and skips unseen words. Word error rate of the baseline is 7.2% and 8.0% for Moroccan and Tunisian respectively. We conducted three sets of experiments:

First, we trained and tested on the same dialectal data. Table 4 (a) shows that we are able to achieve word error rate of 3.1% and 4.0% for Moroccan and Tunisian respectively. When we used Brown clusters (Table 5 (a)), errors decreased by 0.2% absolute for both dialects. In effect, we are able to properly diacritize 56.9% and 50.0% of unseen words or incorrectly diacritized words by the baseline for both dialects respectively.

Second, we wanted to see if both dialects can learn from each other. As Table 4 (b) shows, we trained on one dialect and tested on the other. Adding Brown clusters (Table 5 (b)) lowered results even further. As expected based on our discussion in Section 3., the results were markedly lower, and improvements in diacritizing one dialect would further degrade cross-dialectal results. This validates the claim that word diacritizations in different sub-dialects dialects are significantly different.

Third, we combined training data from both dialects, and we tested on individual dialects. As Table 4 (c) shows, combining data led to results that are worse than the baseline. Using Brown clusters, as shown in Table 5 (c), made results even worse. This is not surprising given the fact that

many words appear in both dialects and are diaritized differently. If both dialects could learn from each other, then perhaps we could have a system that can diacritize either dialect without prior dialect identification. Unfortunately, that is not the case.

6. Discussion and Conclusion

In this paper we presented our work on the diacritization of sub-dialects of Maghrebi Arabic, namely Moroccan and Tunisian. Diacritization is essential for applications such as TTS to properly pronounce words. We noted that dialectal Arabic is less contextual and more predictable than Modern Standard Arabic, and hence high levels of accuracy (low word error rates) can be achieved, to a large extent context free. Using linear chain CRF sequence labeling with character n-grams and character-level Brown clusters, we achieved a word error rate of 2.9% and 3.8% for Moroccan and Tunisian respectively. When we performed cross training the accuracy dropped significantly, which reveals that, even for closely-related dialects, there is a great divergence in pronunciation patterns. For future work, we plan to explore deep learning for diacritization.

7. Bibliographical References

Abandah, G. A., Graves, A., Al-Shagoor, B., Arabiyat, A., Jamour, F., and Al-Taee, M. (2015). Automatic diacritization of arabic text using recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):183–197.

Azmi, A. M. and Almajed, R. S. (2015). A survey of automatic arabic diacritization techniques. *Natural Language Engineering*, 21(03):477–495.

Belinkov, Y. and Glass, J. (2015). Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285, Lisbon, Portugal.

Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Darwish, K., Mubarak, H., and Abdelali, A. (2017). Arabic diacritization: Stats, rules, and hacks. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 9–17.

Eldesouki, M., Samih, Y., Abdelali, A., Attia, M., Mubarak, H., Darwish, K., and Laura, K. (2017). Arabic multi-dialect segmentation: bi-lstm-crf vs. svm. *arXiv preprint arXiv:1708.05891*.

Gal, Y. (2002). An hmm approach to vowel restoration in arabic and hebrew. In *Proceedings of the ACL-02 workshop on Computational approaches to Semitic languages*, pages 1–7. Association for Computational Linguistics.

Habash, N. and Rambow, O. (2007). Arabic diacritization through full morphological tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56. Association for Computational Linguistics.

Habash, N., Eskander, R., and Hawwari, A. (2012). A morphological analyzer for egyptian arabic. In *Proceedings of the twelfth meeting of the special interest group on computational morphology and phonology*, pages 1–9. Association for Computational Linguistics.

Jarrar, M., Habash, N., Alrimawi, F., Akra, D., and Zalmout, N. (2017). Curras: an annotated corpus for the palestinian arabic dialect. *Language Resources and Evaluation*, 51(3):745–775.

Khalifa, S., Hassan, S., and Habash, N. (2017). A morphological analyzer for gulf arabic verbs. *WANLP 2017 (co-located with EACL 2017)*, page 35.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pages 282–289.

Liang, P. (2005). *Semi-supervised learning for natural language*. Ph.D. thesis, Massachusetts Institute of Technology.

Nelken, R. and Shieber, S. M. (2005). Arabic diacritization using weighted finite-state transducers. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 79–86. Association for Computational Linguistics.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT 2013*, pages 380–390. Association for Computational Linguistics.

Rashwan, M., Al Sallab, A., Raafat, M., and Rafea, A. (2015). Deep learning framework with confused sub-set resolution architecture for automatic arabic diacritization. In *IEEE Transactions on Audio, Speech, and Language Processing*, pages 505–516.

Samih, Y., Eldesouki, M., Attia, M., Darwish, K., Abdelali, A., Mubarak, H., and Kallmeyer, L. (2017). Learning from relatives: Unified dialectal arabic segmentation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 432–441.

Zitouni, I., Sorensen, J. S., and Sarikaya, R. (2006). Maximum entropy based restoration of arabic diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 577–584. Association for Computational Linguistics.

Guidelines and Annotation Framework for Arabic Author Profiling

Wajdi Zaghouani, [1] Anis Charfi[2]
[1] College of Humanities and Social Sciences, Hamad Bin Khalifa University, Qatar
[2] Information Systems Program, Carnegie Mellon University Qatar
E-Mail: wzaghouani@hbku.edu.qa, acharfi@qatar.cmu.edu

Abstract

In this paper, we present the annotation pipeline and the guidelines we wrote as part of an effort to create a large manually annotated Arabic author profiling dataset from various social media sources covering 16 Arabic countries and 11 dialectal regions. The target size of the annotated ARAP-Tweet corpus is more than 2.4 million words. We illustrate and summarize our general and dialect-specific guidelines for each of the dialectal regions selected. We also present the annotation framework and logistics. We control the annotation quality frequently by computing the inter-annotator agreement during the annotation process. Finally, we describe the issues encountered during the annotation phase, especially those related to the peculiarities of Arabic dialectal varieties as used in social media.

Keywords: Guidelines, Annotation, Corpus, Arabic, Social media

1. Introduction

Research on author profiling has always been constrained by the limited availability of training data. In fact, collecting textual data with the appropriate meta-data requires significant collection and annotation efforts. For every text, the characteristics of the author have to be known in order to successfully profile the author. Moreover, when the text is written in a dialectal variety such as the Arabic text used in social media, author profiling becomes even more challenging as it requires representative annotated datasets to be available for each dialectal variety.

Arabic dialects are historically related to the classical Arabic and they co-exist with the Modern Standard Arabic (MSA) in a diglossic relation. While standard Arabic has a clearly defined set of orthographic standards, the various Arabic dialects have no official orthographies and a given word could be written in multiple ways in different Arabic dialects as shown in Table 1.

This paper presents the guidelines and annotation work carried out within the Qatar National Research Fund (QNRF) research project on Arabic Author Profiling for Cyber-Security (ARAP)[1]. We used these guidelines in order to create resources and tools for 11 Arabic dialects (Zaghouani and Charfi 2018a; Zaghouani and Charfi 2018b). We collected our ARAP-Tweet corpus data from public Twitter accounts across various regions in the Arab world.

For the author profiling task, most of the currently available resources are for English and other European languages as described by (Celli et al., 2013). The dialectal Arabic resources are still lagging behind other languages when it comes to the availability of the required datasets (Rosso et al., 2018; Zaghouani, 2014).

To the best of our knowledge, there is no dialectal Arabic corpus available for the detection of age, gender, native language and dialectal variety. Having a large amount of annotated data remains the key to reliable results for tasks such as the author profiling. The lack of such resources motivated the creation of the resources presented in this paper.

Once we collected the dataset, we wrote the guidelines for the annotation of Tweets collected for their dialectal variety, their native language, the gender of the user and the age within three categories (under 25 years, 25 to 34 and 35 and above). Furthermore, we hired a team of experienced annotators and we designed an optimized annotation workflow. Moreover, we followed a consistent annotation evaluation protocol in order to validate our annotation protocol.

Variety	Sentence
English	**When I went to the library**
Standard Arabic	عندما ذهبت إلى المكتبة ʿindamā ḏahabtu ʾila l-maktabati
Tunisian	وقتلي مشيت للمكتبة wăqtəlli mʃit l-əl-măktba
Algerian	ملي رحت للمكتبة məlli raḥt l-əl-măktaba
Moroccan	ملي مشيت للمكتبة məlli mʃit lmăktaba
Egyptian	اما رحت المكتبة amma roḥt el-maktaba
Lebanese	لما رحت عالمكتبة lamma reḥit ʕal-mektebe
Iraqi	من رحت للمكتبة min reḥit lil-maktaba
Qatari	لمن رحت المكتبة lamman ruḥt el-maktaba

Table 1: A sample sentence in seven Arabic Dialects

Overall, our corpus has the following features that distinguish it from other Arabic annotation projects:

• **Aim:** designed mainly as a resource for developing Author profiling tools.

[1] http://arap.qatar.cmu.edu/

• **Size:** 2,4 million words.
• **Text types:** Social Media from Twitter

• **Variety:** our data is from 16 Arabic countries representing 11 major Arabic regional dialects.

The remainder of this paper is organized as follows. In Section 2, we discuss related work. Then, we present our ARAP-Tweet corpus collected in Section 3. Section 4 describes our annotation guidelines whereas Section 5 explains our annotation logistics and workflow. Section 6 presents the evaluation of the annotation quality.

2. Related Work

We identified several efforts to create resources for some major Arabic dialects such as Egyptian and Levantine (Diab and Habash, 2007, Pasha et al., 2014, Habash et al., 2013). Within the context of the Qatar Arabic Language Bank (QALB) project, a large-scale annotated corpus of users' comments, the dialectal words were marked and replaced by their equivalent in standard Arabic (Zaghouani et al., 2014; Zaghouani et al., 2015; Zaghouani et al., 2016a.)

In the same context, Salloum and Habash (2013), Sajjad et al. (2013), Salloum and Habash (2013) and Sawaf (2010) used a translation of dialectal Arabic to Standard Arabic as a pivot to translate to English. Zbib et al. (2012) used crowdsourcing approaches to create some resources for machine translation of Arabic dialects.

Al-Sabbagh and Girju (2010) extracted various cues from the Internet to create a lexicon from Dialectal to Modern Standard Arabic. Chiang et al. (2006) built a parser for Dialectal Arabic using the training data from the standard Arabic Penn Treebank. Boujelbane et al. (2013) created a dictionary based on the relation between MSA and Tunisian Arabic.

For the regional dialects, some existing projects were related to dialect identification as mentioned in (Habash et al., 2008; Elfardy and Diab, 2013; Zaidan and Callison-Burch, 2013).

Furthermore, a Twitter dialectal Arabic corpus was created by Mubarak and Darwish (Mubarak and Darwish, 2014) covering four dialectal regions using geolocation information associated with Twitter data.

As the dialectal Arabic is widely used nowadays in most of the informal communication online across the various regions of the Arab world such as in chats, emails, forums and social media, several research efforts were initiated to create dialectal Arabic dedicated tools and resources. However, many of these efforts were disjointed and not coordinated and most of them have only focused on a limited number of dialects or regions that cannot represent the different regions of the Arab world. For instance, some of these resources are not fine-grained with only four major dialectal regions represented such as North Africa, Levant, Egypt, and the Gulf.

For the Arabic author profiling task, the data to be collected is expected to be representative of most of the Arabic dialects and for the moment such resources are not yet available. We found only two projects related to that topic by Abbasi and Chen (2005) and Estival et. al. (2008). The first work focuses on author identification in English and Arabic web forum messages to automatically detect extremist groups. The second work focuses on author profiling for English and Arabic e-mails.

Recently, Bouamor et al. (2018) and Habash et al. (2018) built MADAR and wrote dialectal Arabic unified guidelines to create dialectal Arabic corpus and lexicon covering dialects of various cities across the Arab world with a focus on a travel domain corpus.

For the first time, during the Author Profiling task at PAN 2017 (Rangel et al. 2017),[2] an Arabic task was presented to identify the gender and the dialect using a corpus of four Arabic dialects namely, the North African dialect, the Egyptian Arabic, the Levantine Arabic and the Gulf Arabic. For the resources cited above, the domain was limited in one case and the coverage was limited to only four countries out of 22 Arabic countries in another case.

In our project, we support the major dialects in the Arab world by covering 11 regions and 16 countries. Hence, our project will provide important contributions to Arabic Author profiling.

3. Corpus Description

In this section, we describe the corpus collection and data selection processes carried out to locate and crawl users for each dialect group. For practical reasons, we harvested our data from Twitter as it provides a powerful and free API for crawling and collecting data about public Twitter accounts and public Tweets.

Using the Twitter API and the TweePy[3] library for Python, we collected tweets that contained typical dialectal distinct words generally used by speakers of a given dialect. In other words, we searched for tweets that use dialect specific words and expressions, which allowed us to restrict the tweets to the selected region as much as possible. For example, the word كرهبة /karhba/ 'car' in Tunisian Arabic or the word زول /zo:l/ 'man' in Sudanese Arabic. The seed words for each region were created following a study to identify several seed words for each region. Furthermore, the annotators were trained to identify the cases where a given seed word was used in a profile from another region.

During a six weeks period, we sampled our list of user profiles according to this method. Once we had the initial

[2] http://pan.webis.de/clef17/pan17-web/author-profiling.html
[3] https://github.com/tweepy/tweepy

list of profiles ready for collection, we used Twitter Stream API and the geographic filter to ensure that the collected Tweets are within the specified region. Moreover, we collected the Twitter metadata for each user such as characteristics of the Twitter profile (that are independent of tweet content), to determine demographic information.

As the data collected from social media is usually noisy, we wrote a script to clean the collected Tweets from non-textual content such as images and URLs. Moreover, we filtered all non-Arabic content from the collected data.

For each region, we collected the profiles of 100 users with at least 2000 posted Tweets. For all users, we downloaded up to their last 3240 tweets, which is the limit imposed by Twitter API.

We ended up with a minimum of 200K Tweets per region and a total of 2.4 Million Tweets corpus (Zaghouani and Charfi 2018a).

During the data collection process, we tried to expand our coverage as much as possible taking into consideration the resources and the budget available, we were able to collect a balanced Tweets corpus from 11 Arabic regions representing a total of 16 countries from a total of 22 Arabic countries members of the Arab league as shown in Table 2. We tried to select the data as randomly as possible by avoiding well-known/famous and influential users.

Dialect	Region
Moroccan	1. Morocco
Algerian	2. Algeria
Tunisian	3. Tunisia
Libyan	4. Libya
Egyptian	5. Egypt
Sudanese	6. Sudan
Lebanese	7. North Levant
Syrian	7. North Levant
Jordanian	8. South Levant
Palestinian	8. South Levant
Iraqi	9. Iraq
Qatari	10. Gulf
Kuwaiti	10. Gulf
Emirati	10. Gulf
Saudi	10. Gulf
Yemeni	11. Yemen

Table 2: Dialects and regions selected in the corpus

Once our data is ready, we started a manual annotation step for the collected user profiles in order to: (a) validate the data collected; (b) annotate each user with the age and gender; (c) confirm the dialect used by the users and check if she is a native or non-native speaker of Arabic.

We created general and specific annotation guidelines and we employed a group of annotators to perform the manual annotation for each annotation task.

4. Annotation Guidelines

The annotation guidelines usually document the core of the annotation policy in any given corpus annotation project. Our guidelines are tailored to each of the four annotation tasks within the context of our project: the gender, the age, the dialect and whether the user is a native Arabic speaker or not.

We describe the process of how to annotate each of these tasks, including how to deal with borderline cases. We provided many annotated examples based on our guidelines to illustrate the annotation rules and exceptions for each task. We adopted an iterative approach to develop our guidelines, which includes many revisions and updates as needed in order to reach a consistent set of instructions. For instance, several changes to the guidelines were needed to address the issue of age identification task due to the complexity and the difficulty of this particular task.

The annotations were done by carefully analyzing each of the user's profiles, their tweets, and when possible, we instructed the annotators to use external resources such as personal web pages or blogs as well as other social networks such as LinkedIn and Facebook. We created profiles validation guidelines and task-specific guidelines to annotate the users.

4.1 Profiles Validation Guidelines

To ensure the suitability of the corpus collected for the author profiling task we wrote the annotation guidelines. Moreover, we clearly instructed the annotators on how to validate or exclude the collected Twitter profiles from our data. Finally, we set simple and clear rules and requirements as listed below:

- The profile should be public as we cannot retrieve the data from private or protected profiles.
- The tweets should have been mostly written in the given regional dialect. Moreover, the Tweets should not be mostly written in standard Arabic or any other language such as English or French.
- The profile should represent an actual person (i.e., not a company).
- The profiles posting lots of images and using applications to automatically post daily messages by bots are also filtered out.

4.2 Gender Annotation Guidelines

For some accounts, the annotators were not able to identify the gender as this was based in most of the cases on the name of the person or his/her profile photo and in some cases by their biography or profile description. In case this information is not available, we instructed the annotators to read the user posts and find linguistic indicators of the user's gender.

Like many other languages, Arabic conjugates verbs through numerous prefixes and suffixes and the gender is sometimes clearly marked such as in the case of the verbs ending in taa marbuTa (ة/ـة) which is usually of feminine gender as shown in the example in Table 3.

Form	Sentence
English masc. / fem. Form	*I am thirsty*
Arabic masc. form	أنا عطشان /ana Atshaan/ I am thirsty (Masc.)
Arabic fem. Form	أنا عطشان**ة** /ana atshaan**a/** **I am thirsty (fem.)**

Table 3: Taa Marbuta gender marker in the Arabic verbs

4.3 Age Annotation Guidelines

In order to annotate the users for their age, we used three categories: under 25 years, between 25 years and 34 years, and 35 years and above.

In our guidelines, we asked the annotators to check if the user birth year is available in their Twitter profile. Depending on the dialect region, 4 to 7 % of the users put this information in their public profile. We also asked the annotators to read the latest 100 tweets of the user first for validating their dialect and second for finding any age-related hints. For example, some users had tweets such as "I just turned 25". In some cases, the annotators found some hints indicating that the users were high school or university students such as tweets about exams, schools, and university breaks, etc.

Next, we asked the annotators to retrieve the full name of the user also from their profile and when available search for that name on search engines as well as on other social networks such as LinkedIn and Facebook. The search retrieved for some users their personal homepage or their blog, which could contain their age information. As some Twitter users put their photo in their profile picture this helped the annotators in match twitter users with their respective web page, blog, or social media account on Facebook, Instagram, and LinkedIn. Also other information from the Twitter profile such as the name of the city they live in as well as their job description was helpful for matching accounts on different social networks. In the case of LinkedIn, the graduation year from school or university and also the professional experience were helpful in determining the age group. For example, someone who graduated from university in the year 2000 is certainly above 35 years.

In the last step, if a Twitter profile photo is available the annotators were asked to estimate the age based on that photo (as well as any other photos that the same person may have on their Twitter account in the photos section). Then, we instructed the annotators to use the artificial intelligence based Microsft service How-Old.Net[4], which takes an image in input and determines the subject's age and gender. as shown in Figure. 1. In addition, we wrote a program that automatically retrieves the profiles photos for all selected users and retrieves their age and gender using Microsoft Face API[5]. Even though both tools are from Microsoft they delivered slightly different results.

In the cases, in which age estimation was not possible we replaced the respective Twitter accounts by others of the same gender and from the same region. The newly added accounts were selected so that they provide indications and hints about the age as explained above.

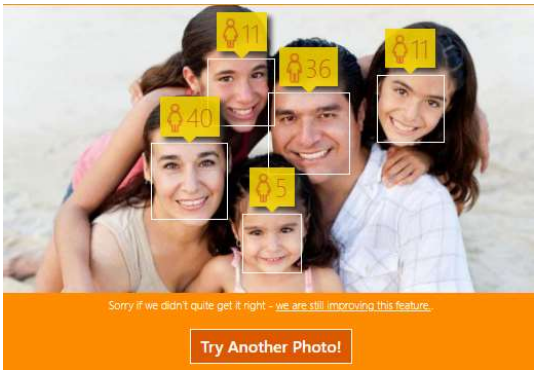

Figure 1: Automatic age estimate sample by How-Old.Net. Photo credit: GSCSNJ (Creative Commons)

4.4 Dialect variety Annotation Guidelines

As the dialect and the regions are known in advance to the annotators, we instructed them to double check and mark the cases in which the user appears to be from a different dialect group. This is possible despite our initial filtering based on distinctive regional keywords. We noticed that in more than 90% the profiles selected belong to the specified dialect group. For the 10% remaining, we observed many cases of people borrowing terms and expressions from other dialects such as in the case of the word بزاف Bizzef 'many' which is typically used in Algerian dialect and also in the Moroccan dialect. In case of doubt, the annotators were instructed to use Google search to check the usage frequency of a given word and to which dialect it is mostly associated.

4.5 Native Language Annotation Guidelines

The goal of this annotation task is to mark and identify Twitter profiles with a native language other than Arabic, so they are considered as Arabic L2 speakers. In order to help the annotators identify those, we instructed them to look for the following cues in order to identify the non-native Arabic users:

- Essays produced by learners of Arabic as second language differ from those of natives, not only quantitatively but also qualitatively. Their

[4] https://how-old.net

[5] https://azure.microsoft.com/en-us/services/cognitive-services/face/

writings display very different frequencies of words, phrases, and structures, with some items overused and others significantly underused.
- Sentences written by Arabic L2 speaker have often a different structure and are not as fluent as sentences produced by a native speaker even when no clear mistakes can be found.
- Style: Arabic L2 Tweets texts may be written in a style that is unfamiliar or unnatural to native speakers although the word order is acceptable, and the sentence conveys the meaning correctly.

Non-native Tweets also contain varying degrees of grammatical, orthographic and lexical errors generally not produced by native speakers. When identifying non-native users, we instructed the annotators to focus on lexical choice errors and syntactic errors as detailed below:

- Word Choice Errors: These include the obvious use of an incorrect word in a given context. Word choice errors are particularly frequent in the L2 Arabic student essays.
- Syntactic Errors: These include a wrong agreement in gender, number, definiteness or case as well as wrong case assignment, wrong tense use, wrong word order.

5. Annotation Logistics

The annotation of a large scale corpus requires the involvement of a team of annotators. In our project, the annotation effort was led by a lead annotation manager who is responsible for the whole annotation task. This includes compiling the data, the annotation of the gold standard Inter-Annotator Agreement (IAA) portion of the corpus, writing the annotation guidelines, hiring and training the annotators, evaluating the quality of the annotation, monitoring and reporting on the annotation progress. To ensure the suitability of the annotators for the various annotation tasks, we selected university level annotators with a good knowledge of the Arabic regional dialects selected. Furthermore, the annotators were screened by doing a limited number of annotation tasks, once hired, they spent a training period of two weeks. During the training period, the annotators read the guidelines, held several group meetings and completed some tasks before starting the official annotation phase.

During the annotation phase and to ensure the quality of the annotated corpus, the annotation manager assigned files to be done by all the annotators and later on, their annotation was compared to compute their Inter-Annotator agreement scores (IAA). Furthermore, a communication message board was provided as space for the annotators to post their questions, add comments, report issues and get feedback from the annotation manager as well as the other annotators. We encouraged the annotators to use this way of communication in order to keep track of all the issues faced and to have an interaction archive that can be used later on to improve the current version of the guidelines.

6. Evaluation

We evaluate the Inter-annotator agreement (IAA) to quantify the extent to which independent annotators, excluding the lead annotator, trained using our guidelines, agree on the annotations. A high level of agreement between the annotators indicates that the annotations are reliable and the guidelines are useful in producing homogeneous and consistent dataset. We created a gold standard dataset of 110 Twitter profiles representing the 11 regions to evaluate the annotators and their application of the guidelines.

During the evaluation, we assigned in a blind way, the sample dataset to all the annotators without any mention to them, so that it was considered as a regular annotation exercise from their end. Later on, we measured the Inter-annotator agreement using Cohen's kappa formula. At the end of the evaluation, we computed the average Kappa scores obtained by the annotators and listed in Table 4. For the gender annotation, they obtained a high score of 0.95, for the age annotation an average score of 0.80, for the dialect identification a score of 0.92 and finally for the native language annotation a relatively low score of 0.70.

As observed, the gender annotation task score was the highest with a near perfect agreement of 95%. For the dialect identification task, some annotators were confused by a few similarities that exists between some dialects such as the Moroccan dialect and the Algerian dialect and also by the Qatari dialect and some other Gulf dialects.

The age identification task proved to be a difficult task, especially with the absence of clear cues and indicators such the birth year, graduation year and the absence of a profile photo.

Finally, the native language identification ranked last as it could be very hard to find due to the lack of cues. Overall, we believe that the annotation agreement is above the acceptable range given the difficulty of the tasks.

Task	Kappa Score
Gender Annotation	**0.95**
Dialect Annotation	**0.92**
Age Annotation	**0.80**
Native Language	**0.70**

Table 4: Inter-annotator agreement in terms of average Kappa score; the higher the better

7. Conclusion

We presented a set of guidelines and our annotation pipeline to build a large 2.4M annotated Tweets Arabic author profiling corpus called ARAP-Tweet. We summarized our general and dialect-specific guidelines for each of the 11 Arabic dialectal regions collected. The guidelines and the resource created could be used for tasks other than author profiling. In the future, we plan to release the guidelines and the corpus[6] to the research community during the 3rd Workshop on Open-Source Arabic Corpora

[6] As per the Twitter agreement and policy and in order to protect the privacy of the users, we will only distribute the Tweet IDs in the public data release.

and Processing Tools.[7] Moreover, the corpus will be provided to the participants of the author profiling task during the 18th evaluation lab on digital text forensics, PAN @ CLEF 2018.[8]

9. Acknowledgements

This publication was made possible by NPRP grant 9-175-1-033 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

8. Bibliographical References

Abbasi A. and Chen H. (2005) Applying authorship analysis to extremistgroup web forum messages. IEEE Intelligent Systems, 20(5): 67–75.

Al-Sabbagh, Rania and Girju, Roxana. (2010). Mining the Web for the Induction of a Dialectical Arabic Lexicon. In LREC, Valetta, Malta.

Boujelbane, Rahma, Ellouze Khemekhem, Mariem, and Belguith, Lamia Hadrich. (2013). Mapping Rules for Building a Tunisian Dialect Lexicon and Generating Corpora. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, pages 419–428, Nagoya, Japan.

Celli, F., Pianesi, F., Stillwell, D., and Kosinski, M. (2013). Workshop on computational personality recognition (shared task). In Proceedings of the Workshop on Computational Personality Recognition, Boston, MA, USA

Chiang, David, Diab, Mona, Habash, Nizar, Rambow, Owen, and Shareef, Safiullah. (2006). Parsing Arabic Dialects. In Proceedings of EACL, Trento, Italy.

Diab, Mona and Habash, Nizar. (2007). Arabic Dialect Processing Tutorial. In NAACL.

Elfardy, Heba and Diab, Mona. (2013). Sentence Level Dialect Identification in Arabic. In Proceedings of the Association for Computational Linguistics, pages 456–461, Sofia, Bulgaria.

Estival, D., Gaustad, T., Hutchinson, B., Pham, S. B., and Radford, W. (2008): Author Profiling for English and Arabic Emails. Natural Language Engineering, Cambridge University Press (2008).

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

Habash, N. Y. (2010). Introduction to Arabic natural language processing, volume 3. Morgan & Claypool Publishers.

Habash, Nizar, Roth, Ryan, Rambow, Owen, Eskander, Ramy, and Tomeh, Nadi. (2013). Morphological analysis and disambiguation for dialectal arabic. In Proceedings of NAACL-HLT, pages 426–432, Atlanta, Georgia, June.

Pasha, Arfath, Al-Badrashiny, Mohamed, Kholy, Ahmed El, Eskander, Ramy, Diab, Mona, Habash, Nizar, Pooleery, Manoj, Rambow, Owen, and Roth, Ryan. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In Proceedings of LREC, Reykjavik, Iceland.

Rangel F, Rosso P, Potthast M, Stein B (2017) Overview of the 5th Author Profiling Task at PAN 2017: Gender and Language Variety Identification in Twitter. In Proceedings of PAN 2017, Dublin, Ireland

Rosso P, Rangel F, Hernández Farías I, Cagnina L, Zaghouani, W, Charfi A. (2018). A survey on author profiling, deception, and irony detection for the Arabic language. Lang Linguist Compass.; https://doi.org/10.1111/lnc3.12275

Sajjad, Hassan, Darwish, Kareem, and Belinkov, Yonatan. (2013). Translating dialectal Arabic to English. In Proceedings of ACL, Sofia, Bulgaria.

Salloum Wael and Nizar Habash. 2013. Dialectal Arabic to English Machine Translation: Pivoting through Modern Standard Arabic (2013). In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Atlanta, GA.

Sawaf Hassan. (2010). Arabic dialect handling in hybrid machine translation. In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), Denver, Colorado

Zaidan, Omar and Callison-Burch, Chris. (2013). Arabic dialect identification. Computational Linguistics.

Zbib, Rabih, Malchiodi, Erika, Devlin, Jacob, Stallard, David, Matsoukas, Spyros, Schwartz, Richard, Makhoul, John, Zaidan, Omar F., and CallisonBurch, Chris. (2012). Machine translation of Arabic dialects. In Proceedings of NAACL-HLT, Montréal, Canada.

Zaghouani, Wajdi (2014), Critical survey of the freely available Arabic corpora. In Proceedings of the 1st Workshop on Open-Source Arabic Corpora and Processing Tools, 9th International Conference on Language Resources and Evaluation, Reykjavik, Iceland.

9. Language Resource References

Bouamor, Houda, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann and Kemal Oflazer. (2018). The MADAR Arabic Dialect Corpus and Lexicon. In Proceedings of the 11th International Conference on Language

[7] http://edinburghnlp.inf.ed.ac.uk/workshops/OSACT3/

[8] http://pan.webis.de/clef18/pan18-web/index.html

Resources and Evaluation.

Habash, Nizar, Salam Khalifa, Fadhl Eryani, Owen Rambow, Dana Abdulrahim, Alexander Erdmann, Reem Faraj, Wajdi Zaghouani, Houda Bouamor, Nasser Zalmout, Sara Hassan, Faisal Al shargi, Sakhar Alkhereyf, Basma Abdulkareem, Ramy Eskander, Mohammad Salameh and Hind Saddiki. (2018), Unified Guidelines and Resources for Arabic Dialect Orthography. In Proceedings of the 11th International Conference on Language Resources and Evaluation.

Mubarak, H., & Darwish, K. (2014). Using Twitter to collect a multi-dialectal corpus of Arabic. In Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP) (pp. 1-7).

Zaghouani, Wajdi and Charfi, Anis. (2018a). ArapTweet: A Large Multi-Dialect Twitter Corpus for Gender, Age and Language Variety Identification. In Proceedings of the 11th International Conference on Language Resources and Evaluation

Zaghouani, Wajdi and Charfi Anis. (2018b). The ArapTweet Corpus version 0.8. Carnegie Mellon University Qatar, Doha, Qatar

*i*ArabicWeb16: Making a Large Web Collection More Accessible for Research

Khaled Yasser, Reem Suwaileh, Abdelrahman Shouman, Yassmine Barkallah, Mucahid Kutlu, Tamer Elsayed
Computer Science and Engineering Department, Qatar University, Doha, Qatar
{khaled.yasser, reem.suwaileh, a.shouman, yassmine.barkallah, mucahidkutlu, telsayed}@qu.edu.qa

Abstract

ArabicWeb16 is the largest publicly-available Arabic Web crawl, containing 150M Web pages. We envision many uses of this dataset to advance the research in various fields such as information retrieval (IR), natural language processing, and machine learning. However, accessing such a large dataset needs high storage and processing resources, which may not be available for many research teams. In this paper, we present *i*ArabicWeb16, a freely-available Web-based tool making ArabicWeb16 dataset more accessible to the research community via both Web interface and programming API. *i*ArabicWeb16 allows users (typically researchers) to search ArabicWeb16 efficiently while providing them with various ranking methods, besides the ability to download resulting Web pages directly. We evaluate its efficiency and scalability with respect to the number of users it can serve, and show that it is a valuable tool that helps researchers explore and search ArabicWeb16 dataset for the sake of their research work without the storage and computational burden.

1. Introduction

Arabic is one of the most commonly spoken languages in the world with more than 400M speakers, many of whom search the Web daily, making research on Arabic Information Retrieval (IR) an important area. However, research on Arabic IR has been obstructed by the lack of available resources, e.g., datasets, tools, and test collections, that ease the development of retrieval systems.

There are a number of studies developing Arabic datasets and test collections for different IR tasks and domains. Two examples are the test collection of MSA news articles from Agence France Press (AFP) introduced by the TREC Cross-Lingual Track (Gey and Oard, 2001), and EveTAR test collection that consists of 390M Arabic tweets of which 62K are annotated for multiple IR tasks such as event detection and real-time summarization (Hasanain et al., 2017). While these datasets and test collections are useful, researchers may encounter several challenges in developing their prototypes due to limited tools and libraries for Arabic IR systems. A few Arabic search engines are built for either commercial purposes or experimental research (Al-Maimani et al., 2011) such as Idrisi[1], Sawafi[2], and Barq (Rachidi et al., 2003); however they are either too old to the nature and scale of the current Web, or even no longer available.

Recently, Suwaileh et al. (Suwaileh et al., 2016) introduced *ArabicWeb16*, the largest available Arabic Web crawl of 150M Web pages. The collection has the potential to be a valuable resource that can advance the Arabic IR (and related areas such as natural language processing) research in several directions. Many researchers, however, may find it challenging to access, process, and search a collection of that scale. Dataset-specific search and lookup tools were previously introduced to help explore and analyze large crawls such as ClueWeb09 and Clueweb12[3]. Furthermore, the Microblog track in TREC 2013 (Lin and Efron, 2013) and 2014 (Lin et al., 2014) provided a common API by which users can search the large tweet collections. Nevertheless, all of those tools focus only on English data.

In this paper, we present *i*ArabicWeb16, a research tool that provides search and lookup services designed specifically for ArabicWeb16. Processing a dataset in the scale of 150M Web pages brings many computational challenges, requiring high storage resources and computation power. Therefore, to make it more accessible to a wide-range of researchers, we built a search interface front-end that is supported by a customizable Lucene-based back-end. The back-end runs in a multi-threaded fashion to speed up the search process, while the front-end allows users to try various ranking functions (e.g., language-modeling and BM25 (Robertson et al., 1995)) and set search fields (e.g., title vs. content) among other features. This flexibility allows users to set the best configurations for their search needs more effectively. Moreover, users can retrieve the content of the documents directly using their document IDs, which further helps users needing only a subset of the crawl. Finally, *i*ArabicWeb16 provides access to ArabicWeb16 via both Web interface and programming API.

We evaluated the efficiency of *i*ArabicWeb16 in various situations. Our experiments show that it returns search results for a single user in less than 200ms on average by employing 64 threads, and can also serve 128 users (the largest set we tested) concurrently without a huge delay (with average response time of 6.5 seconds).

2. ArabicWeb16 Collection

ArabicWeb16 is a one-month snapshot (1st-30th January 2016) of the Arabic Web that contains around 150M Web pages (Suwaileh et al., 2016). At the time of this writing, ArabicWeb16 is the largest Arabic Web dataset which is publicly available for the research community. The dataset contains diverse types of Web pages such as informational pages (e.g., Wikipedia), forums, news articles, organizational pages, and transactional pages. It also covers high linguistic diversity of Arabic by containing around 30M web pages with different Arabic dialects. We believe the dataset can be a useful resource for many research areas such as machine learning, natural language processing, and

[1] www.aramedia.com/idrisi.htm
[2] www.multilingual-search.com/sawafi-a-new-arabic-search-engine/
[3] lemurproject.org

IR. For IR, it can be used for research on search tasks (e.g., ad-hoc Web and blog search, cross-dialect search), filtering (e.g., news), question-answering (e.g., over blogs and forums), and spam detection among others.

3. *i*ArabicWeb16 Architecture

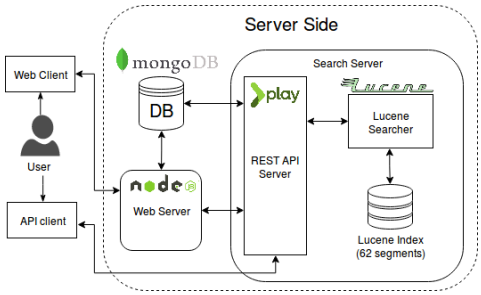

Figure 1: *i*ArabicWeb16 Architecture.

In this section, we present the architecture of *i*ArabicWeb16 in detail. It has three main components (see Figure 1):

- **Web Server**: Provides a Web interface for users to perform search tasks and use other functionalities of the system, e.g., requesting API keys.
- **Search Server**: Performs search tasks over the collection and retrieves documents. It consists of three components (REST server, Lucene Searcher, and Lucene Index) to handle the search tasks.
- **Database**: Stores the raw documents (in HTML format) to be displayed when documents are retrieved.

A typical search scenario on *i*ArabicWeb16 is as follows. Only a user with a valid API key can access the search server and perform search operation through the API or the Web interface. A search query submitted by a Web client (i.e., a user using the Web interface) or an API client (i.e., a user using the API) is first processed by the REST Server to make it ready for search operations. Next, Lucene Searcher performs the search task over the index using the search parameters that the user provides. Subsequently, the REST Server performs a post-processing (i.e., filtering) to have a better representation of the results. Finally, the results are retrieved from the database and returned to the user.
We next discuss the back-end (Section 3.1.) and the front-end (Section 3.2.) in more details.

3.1. Back-End

In this section, we explain some implementation details of the back-end and how search requests are handled.

3.1.1. Storing HTML Pages

Fast retrieval of documents is significant for the overall performance of the system. Therefore, we store the raw HTML content of the Web pages in a database to be able to present the results in the Web interface. We used MongoDB 3.2 [4] in which the field size limit is large enough to contain large Web pages.

[4] www.mongodb.com/

3.1.2. Indexing

To prepare for search, we indexed ArabicWeb16 collection using Lucene 6.2.[5]. Indexing the entire collection would take a very long time on a single thread; therefore, we partitioned the dataset into 31 partitions (equal to the number of folders of the raw collection) and indexed each separately in parallel. We then merged them to form the final index. Furthermore, the index contains both stemmed and non-stemmed versions of each document, allowing users to enable/disable stemming for each search query. We used the default Arabic stemmer of Lucene for stemming.
The total size of the index is about 1.6TB, bringing additional challenges to efficient search. To improve the efficiency of the search tasks, we split the index into 62 segments allowing multiple search threads to run more efficiently.

3.1.3. Searching

*i*ArabicWeb16 is designed to help researchers work on Arabic IR. Therefore, instead of implementing a static search engine with fixed configurations, we developed a configurable search engine in which several settings can be chosen by the researchers. In fact, they can explore different ranking algorithms, search fields, and indexes. The Lucene Searcher provides 5 different ranking functions: TF-IDF, BM25 (Robertson et al., 1995), Query-Likelihood with Dirichlet smoothing (Ponte and Croft, 1998), Query-Likelihood with Jelinek-Merce smoothing (Zhai and Lafferty, 2004), and combination of all using CombSUM method (Shaw et al., 1994). It also performs search on title only, content only, or both title and content, with stemming on or off. Finally, the number of returned results can also be specified.

3.1.4. Processing Search Requests

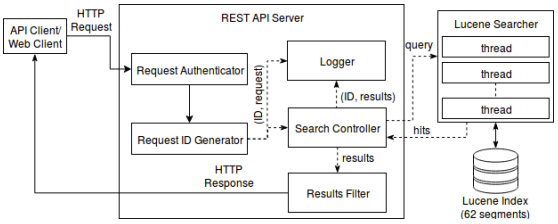

Figure 2: The process of handling search requests. Dashed lines indicate asynchronous calls.

Figure 2 illustrates the process of handling search requests. Search requests from both API and Web clients are sent as HTTP requests to the search server, which uses Play Framework 2.6[6] to implement its Web services. Once an HTTP request is received, its API key needs to be authenticated. A unique ID is then generated and the request itself is logged and sent to the *Search Controller* (SC). SC performs the required pre-processing on the query (e.g., stemming if set) and issues the search query against the multi-threaded Lucene Searcher (LS). Each thread of the LS runs on a single index segment at a time. Once the search operations are

[5] lucene.apache.org/
[6] playframework.com/

Figure 3: Search interface: an example of searching ArabicWeb16. The translation of the search options are: (1) Search bar (query: tourism trips in Azerbaijan), (2) The number of results, (3) Save selected results, (4) Save all results, (5) Show/hide advanced search options, (6) Ranking function, (7) Search field, (8) Show snippets, and (9) Stemming.

done for all segments, the search results are returned to SC, which passes the results to the *Results Filter* (RF). RF removes the duplicates based on their content and groups the results from the same source to have a better presentation of the search results. After filtering, the results are sent back to the client in an HTTP response.

Our system handles multiple search queries concurrently. In order to avoid a request being delayed because another one is being processed, all internal calls are performed asynchronously such that components do not wait for one another to handle the next request. The asynchronous calls are shown in dashed lines in Figure 2.

As users can also request specific documents by IDs, processing document retrieval requests is done in a similar way, but with two differences. First, the database is queried instead of the Lucene Searcher. Second, RF compresses the documents if multiple are requested.

3.2. Front-End

*i*ArabicWeb16 allows users to search through the Web interface or the programming API. In this section, we describe each of the two ways and provide examples on how to use them.

3.2.1. Search Interface

*i*ArabicWeb16 provides a Web search interface[7] that allows the registered users to perform interactive search similar to commercial search engines. Figure 3 shows a simple search task using the search interface (search options are translated below the figure for convenience). The interface reflects the options provided by the back-end searcher (discussed in Section 3.1.3.) via specifying the number of returned results, the ranking function, the search fields, and enabling/disabling word stemming. The user can also choose to display snippets of results. Once results are returned, users can click on each result to see either the crawled version of the page or the live (current) one.

The default search options are set as follows. The number of results is set to 10; the ranking method is set to Query-Likelihood (Dirichlet); search field is set to content-only; and both stemming and snippets are enabled.

3.2.2. Programming API

*i*ArabicWeb16 also provides a Java client API which enables developers or researchers to perform search operations with different configurations and retrieve documents directly using their IDs within their programs. Figure 4 shows the signatures of the most important functions provided by the API.

- **`search`**: enables the users to issue a specific query on the collection with the specified configuration (e.g., ranking function, number of returned results, etc.). It returns a string in JSON format that can be parsed to an array of results using **`parseResults`** function (not shown in the figure).
- **`retrieveSingleDoc`**: returns the document with the given ID.

[7] `bigir1.qu.edu.qa:3000`

- **retrieveBatchOfDocs**: writes the content of the requested documents in the destination file specified by the user in a compressed format.

```
String search(query, configuration)
String retrieveSingleDoc(docID)
void retrieveBatchOfDocs(docID[],
    destinationFile)
```

Figure 4: Example API functions.

4. Performance Evaluation

In this section, we report experiments that we conducted to evaluate efficiency and scalability of *i*ArabicWeb16.

4.1. Experimental Setup

We host the search server and conducted our experiments on an Oracle Linux 7.4 server with 128 GB memory and 2 Intel Xeon E5-2660 v3 2.6 GHz CPUs having 40 cores in total (20x2). In order to simulate users, we used a set of 7052 queries used for obtaining seed pages when crawling ArabicWeb16.

4.2. Search Response Time

In this experiment, we measure the response time with respect to the number of threads used for search tasks. Specifically, we vary the number of search threads from 2 to 64, and in each case, we issue 500 sequential search queries sampled from our set from a single user (i.e., no concurrent search requests). The search queries are selected randomly and the same query set is used for each case. For each case, we compute the average search response time. The results are depicted in Figure 5. The vertical bars represent the standard deviation across the queries for each case. The Figure shows that the response time decreases by a factor of 5 when the number of threads increases 32 times. This speedup is due to the fact that LS is capable of searching many segments in parallel. Note that, although we conducted an experiment at 64 threads, the final deployed server can use up to 40 threads because the server machine we used has a total of 40 cores.

4.3. Scalability

In this experiment, we test the scalability of *i*ArabicWeb16. We vary the number of users using the search engine concurrently from 2 to 128, doubling in each case. For each user, we run 10 queries randomly selected from our query pool and compute the average response time over the 10 queries. We then compute the mean of average response time of clients for each case. The results are shown in Figure 6. When we increase the number of users by 64 times, the response time increases by a factor of 60 times, reaching an average of 6.5 seconds when 128 users are issuing queries in parallel to the server.

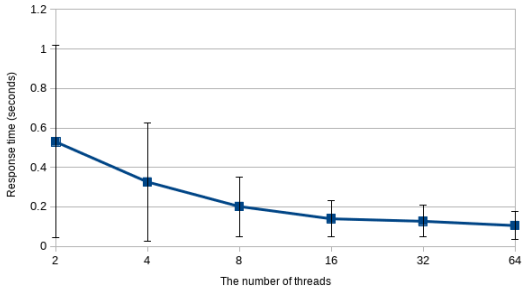

Figure 5: The average response time with varying number of search threads. The vertical bars represent the standard deviation across queries.

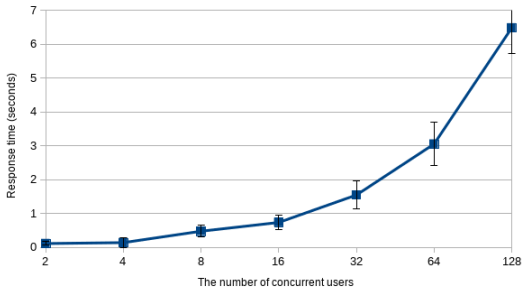

Figure 6: The response time of the server as the number of concurrent users increases. The error bars represent the average standard deviation across the clients.

5. Conclusion and Future Work

In this paper, we present *i*ArabicWeb16, a search service that makes ArabicWeb16, the largest publicly available Arabic Web crawl, more accessible to the research community. Using *i*ArabicWeb16, researchers can search over ArabicWeb16 using the Web interface and the programming API. *i*ArabicWeb16 provides a flexible search service in which users can choose different ranking functions and search fields, and can get access to the content of the retrieved Web pages. Our experiments showed that *i*ArabicWeb16 is an efficient research tool and can serve multiple users at the same time with a reasonable response time; therefore, it can be used by many researchers who would like to use ArabicWeb16 in their research.

In the future, we plan to deploy the search engine in a distributed environment to further increase its efficiency. We also plan to extend *i*ArabicWeb16 to develop search topics and collect relevance judgments in order to help researchers construct their own test collection over ArabicWeb16.

Acknowledgments

This work was made possible by NPRP grant# NPRP 7-1313-1-245 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Al-Maimani, M. R., Al Naamany, A., and Bakar, A. Z. A. (2011). Arabic information retrieval: techniques, tools and challenges. In *GCC Conference and Exhibition (GCC), 2011 IEEE*, pages 541–544. IEEE.

Gey, F. C. and Oard, D. W. (2001). The TREC-2001 cross-language information retrieval track: Searching arabic using english, french or arabic queries. In *TREC*, pages 16–26.

Hasanain, M., Suwaileh, R., Elsayed, T., Kutlu, M., and Almerekhi, H. (2017). EveTAR: Building a large-scale multi-task test collection over Arabic tweets. *Information Retrieval Journal*, pages 1–30.

Lin, J. and Efron, M. (2013). Overview of the TREC-2013 microblog track. In *Proceedings of the 22nd Text REtrieval Conference*, TREC '13.

Lin, J., Efron, M., Wang, Y., and Sherman, G. (2014). Overview of the TREC-2014 microblog track. In *Proceedings of the 23rd Text REtrieval Conference*, TREC '14.

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM.

Rachidi, T., Bouzoubaa, M., ElMortaji, L., Boussouab, B., and Bensaid, A. (2003). Arabic user search query correction and expansion. *Proc. of COPSTIC*, 3:11–13.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., Gatford, M., et al. (1995). Okapi at TREC-3. *Nist Special Publication Sp*, 109:109.

Shaw, J. A., Fox, E. A., Shaw, J. A., and Fox, E. A. (1994). Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2*, pages 243–252.

Suwaileh, R., Kutlu, M., Fathima, N., Elsayed, T., and Lease, M. (2016). ArabicWeb16: A new crawl for today's Arabic web. In *Proceedings of the 39th International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 673–676. ACM.

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214.

Learning Subjective Language: Feature Engineered vs. Deep Models

Muhammad Abdul-Mageed
Natural Language Processing Lab
University of British Columbia
muhammad.mageeed@ubc.ca

Abstract

Treatment of subjective language is a vital component of a sentiment analysis system. However, detection of subjectivity (i.e., subjective vs. objective content) has attracted far less attention than sentiment recognition (i.s., positive vs. negative language). Particularly, online social context and the structural attributes of communication therein promise to help improve learning of subjective language. In this work, we describe successful models exploiting a rich and comprehensive feature set based on the structural and social context of the Twitter domain. In light of the recent successes of deep learning models, we also effectively experiment with deep gated recurrent neural networks (GRU) on the task. Our models exploiting structure and social context with an SVM achieve $> 12\%$ accuracy higher than a competitive baseline on a blind test set. Our GRU model yields even better performance, reaching 77.19 (i.e., $\sim 14.50\%$ higher than the baseline on the same test set, $p < 0.001$).

1. Introduction

Ability to detect subjective language (i.e., aspects of language expressing opinions, feelings, evaluations, and speculations (Banfield, 1982)) is an important part of any real-world sentiment system where a unit of analysis is usually labeled as either objective (e.g., *I read the book.*) or subjective. Subjective texts are further classified into sentiment categories as *positive* (e.g., *This market is spectacular!*), *negative* (e.g., *This machine is unfortunately very slow.*), or *mixed* (e.g., *The new models are powerful, but quite memory-intensive!*). In spite of an excellent (early) thread of literature targeting learning subjective language that focused on utilizing lexical and syntactic information (Wiebe et al., 2004; Wilson et al., 2006)), gender (Burger et al., 2011; Rao et al., 2010; Volkova et al., 2013; Volkova et al., 2015), and discourse features (e.g., punctuation, emoticons) (Benamara et al., 2011), the field has focused more on sentiment or polarity classification rather than subjectivity.

Particularly social media communication takes place in a very different, yet rich, context: First, Twitter language diverges from the 'standard' offline language in various *structural* ways. For example, Twitter tweets are a maximum of 140 characters per tweet. Twitter is also an environment where users re-tweet other users, address them using an '@' sign, tag tweets and/or launch tweet campaigns using hashtags, share URLs, etc. Rather than viewing these unique structural attributes of the Twitter domain as challenges, we seek to exploit them for learning subjectivity in the context of the microblogging platform.

Second, communication on Twitter happens against its wider *social* context where user identities, gender, race, age, economic class, etc. are all attributes that afford cues which can be leveraged for social meaning extraction tasks like that of subjectivity. Although (at times scattered) features based on the structure of Twitter language and its social context have been used in the literature, a unified and systematic analysis of the collection of *structural and social context* features that can inspire further work in the field, especially for the Arabic language, is lacking. As such, we describe novel and successful models exploiting a rich feature set (totaling 30 features thematically organized in 11 feature groups) based on the structural and social attributes of the the Twitter domain. Examples of *structural* features we employ include use of hashtags, non-standard typography (e.g., letter repetition, use of emoticons), and use of URLs. Instances of *social* features we leverage include user id and user gender. We provide a more detailed account of our feature set in Section 4.1..

Third, while there are several methods for feature selection, including for text classification (e.g., (Dash and Liu, 1997; Yang and Pedersen, 1997; Forman, 2003; Chandrashekar and Sahin, 2014)), finding the relevant features and the best combinations of these from a feature set composed of a large number of features can be challenging, if not impossible. In this work, we introduce two methods of feature selection aimed at identifying the best performing feature combinations from among the 30 proposed features.

Finally, deep learning of natural language (LeCun et al., 2015; Goodfellow et al., 2016; Goldberg, 2016) has shown impressive successes in recent years. It is yet unknown, however, to what extent a deep learning system would compare to a system based on careful feature engineering using domain knowledge of the type provided in this work in the context of subjectivity classification. Our work here seeks to at least partially bridge this gap by comparing a feature-engineered system to a carefully-designed deep learning system tackling the problem.

Overall, we make the following contributions: (1) We propose a rich set of structural and social context features that we exploit for learning subjective language onnline (i.e., on a Twitter dataset), (2) We describe two feature selection methods that enable a semi-exhaustive search for the best feature combinations from a large number of features that are otherwise hard to search, and (3) We develop a highly effective gated recurrent neural network model for the task, showing the utility of this class of methods and how it is that these compare to our expertly hand-crafted system exploiting the features we introduce.

The rest of the paper is organized as follows: In section 2., we discuss related work. In Section 3., we describe our dataset. In Section 4., we describe our models with hand-crafted features. In Section 6., we introduce our model based on gated recurrent neural networks and present its results acquired with it. In Section, 7. we conclude.

2. Related Work

Subjectivity in Natural Language *Subjectivity* in human language, as introduced earlier, refers to aspects that express opinions, feelings, evaluations, and speculations (Banfield, 1982). There is a vast literature on subjectivity and sentiment analysis. Early computational treatment of subjectivity (e.g., (Wiebe, 2000; Wilson et al., 2006)) focused on the lexical and syntactic cues characterizing subjective texts. Our work differs in that we utilize structural and social context features. More recent works investigate exploiting demographic features of the type we incorporate in our feature set here. Especially gender has received significant attention as an attribute that correlates with subjective language (Burger et al., 2011; Rao et al., 2010; Volkova et al., 2013; Volkova et al., 2015). Discourse features, including punctuation- and emoticon-based features, have also been studied in the context of improving subjectivity detection (Benamara et al., 2011).

A number of social context features have also been used for predicting phenomena related to subjectivity. For example, (Persing and Ng, 2014) employ information related to political orientation, relationship status, and health behavior (e.g., drinking, smoking) to predict voting from comments posted on a polling social platform. Similarly, (Thomas et al., 2006) report benefiting from user mentions (e.g., using the "@") network for predicting votes and (Tan et al., 2011) acquire enhanced sentiment classification by incorporating the Twitter follower/followee and user mentions network. (Hasan and Ng, 2013) incorporate sequential user interactions and ideological orientation in debate web fora for stance detection. (Deng et al., 2014) similarly use network-based information between users to improve sentiment classification both at the post and user levels. (Ren et al., 2016) embed user tweets and topics in a neural framework for improving Twitter sentiment analysis. Likewise, a number of researchers, e.g., (Mohammad and Kiritchenko, 2015; Purver and Battersby, 2012; Wang et al., 2012) makes use of Twitter hashtags as a way to automatically label data for the related task of emotion detection, while a string of works considers textual clues (e.g., negation, epistemic modality) interacting with subjective language (Wiegand et al., 2010; Kennedy and Inkpen, 2006). Our work is is similar in that we exploit a wide range of these features, while expanding them and proposing methods enabling searching for their best combinations in the context of classification.

For modeling the related task of sentiment, researchers have typically exploited lexical features using simple frequency statistics of input text (Wiebe, 2000; Wiebe et al., 2004), or modeling the semantics of certain word categories, e.g., dynamic and gradable adjectives (Hatzivassiloglou and Wiebe, 2000) or different semantic classes of verbs (Benamara et al., 2007; Breck et al., 2007)).

A considerable body of the literature has focused on developing or learning polarity lexica (Lin and Hauptmann, 2006; Baccianella et al., 2010; Turney, 2002). Other works have exploited syntactic features like part of speech tags (Gamon, 2004; Hatzivassiloglou and McKeown, 1997) and different *N*-gram windows as a measure to capture (potentially syntactic) context beyond single words (Ng et al., 2006; Cui et al., 2006), syntactic constituents, e.g., (Klenner et al., 2009; Wilson et al., 2005), dependency parses, e.g., (Kessler and Nicolov, 2009; Zhuang et al., 2006; Ng et al., 2006), etc. A few studies have focused on languages of rich morphology, including (Abdul-Mageed et al., 2014) who built systems using gold-processed, treebank data exploiting mophosyntactic information. Other works on Arabic involved building resources that were then used for developing models primarily based on N-gram features (Aly and Atiya, 2013; ElSahar and El-Beltagy, 2015; Mourad and Darwish, 2013) or sub-word information (Abdul-Mageed, 2017b; Abdul-Mageed, 2017a). Some works have focused on modeling dialects (Abdul-Mageed et al., 2014), or the related task of emotion (Abdul-Mageed et al., 2016), yet these remain relatively limited. Recent efforts to collect large-scale Arabic dialectal corpora promise to aid dialect-specific sentiment research (Abdul-Mageed et al., 2018). The focus of our work is different in that we target structural and social features.

Deep Learning Models An increasingly growing number of studies have applied deep neural networks to the problem of sentiment analysis. These include, e.g., (Labutov and Lipson, 2013; Maas et al., 2011; Tang et al., 2014b; Tang et al., 2014a) who learn sentiment-specific word embeddings (Bengio et al., 2003; Mikolov et al., 2013) from neighboring text. Some studies have focused on learning semantic composition (Mitchell and Lapata, 2010; Socher et al., 2013; Irsoy and Cardie, 2014; Li et al., 2015; Le and Mikolov, 2014; Tang et al., 2015) for modeling sentiment. Long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Neural Nets (GRUs) (Cho et al., 2014; Chung et al., 2015), variations of recurrent neural networks (RNNs) have also been used successfully for sentiment analysis (Ren et al., 2016; Liu et al., 2015; Tai et al., 2015; Tang et al., 2015; Zhang et al., 2016). Convolutional neural networks (CNNs) have also been quite successful, including on sentiment analysis (Blunsom et al., 2014; Kim, 2014; Zhang et al., 2015). A review of neural network methods for NLP can be found in (Goldberg, 2016). Our work is similar to these works in that we use GRUs for learning subjective language, basically a text classification task.

For a more comprehensive background on modeling subjectivity and sentiment, readers can refer to a number of excellent comprehensive overviews, including (Pang and Lee, 2008), (Liu, 2012), and (Montoyo et al., 2012). In addition, (Benamara et al., 2017) provide a more recent thorough review of various aspects of evaluative text, including some aspects of social context [1] (e.g., social network structure and user profiles).

[1](Benamara et al., 2017) use the term 'extra-linguistic information' to refer to what we call *social context* in this paper.

3. Data Set and Annotation

For this work, we collect a corpus of $3,015$ Arabic Tweets from the public Twitter timeline and task two college-educated native speakers of Arabic on labeling the data after providing detailed annotation instructions and training as described in (Abdul-Mageed et al., 2014). The data were manually inspected for possible duplicates before we shared with the annotators, and so the $3,015$ are unique. Table 1 shows the the distribution of the SSA categories over the data. As Table 1 shows, 47.36% of the data are assigned an objective (OBJ) tag and the remaining 52.64% has one of the various subjective tags: Positive (S-POS), negative (S-NEG), and mixed (S-MIXED). [2] Inter-annotator agreement on the data reached a Cohen (Cohen, 1996) Kappa *(K)*$= 85\%$. We take the labels assigned by a random judge from among the two annotators to be our gold standard. As Arabic is known to have multiple dialects in addition to its modern standard variety, we also ask annotators to assign each tweet a tag representing whether the variety is Modern Standard Arabic (MSA) or dialectal Arabic (DA). The MSA of the corpus comprises $1,466$ ($\% = 48.62$) tweets, and the dialectal part comprises $1,549$ ($\% = 51.38$) tweets. We do not exploit these language variety tags for this work.

4. Models Based on Hand-Crafted Features with Support Vector Machines

We both use a classical machine learning classifier, introduced here, and a deep learning classifier, which we will introduce in Section 6.. For our models with hand-crafted features, we use an SVM (Vapnik, 1995) classifier with a linear kernel. SVMs are know to perform well on text classification (Joachims, 2002), especially with carefully-designed feature sets. We now turn to introducing our gated recurrent neural network model.

4.1. Features

In order to exploit the structural and social context, we introduce a very rich feature set composed of a total of 30 features. To facilitate reference, we divide these features thematically (with as much coherence per group as is possible) into 11 groups. Although the features target the Twitter domain, we believe they can also be exploited for other domains like chat fora. Our features are inspired by research within the sentiment literature, but also by related areas such as stance detection (Hasan and Ng, 2013), voting prediction (Thomas et al., 2006; Persing and Ng, 2014), and social media and computer-mediated communication (Herring, 2007; Androutsopoulos and Beißwenger, 2008; Herring et al., 2013; Bieswanger, 2013). We now describe our feature set.

User Gender: Inspired by gender variation research exploiting social media data, e.g., (Herring, 1994), and previous research on sentiment analysis (Volkova et al., 2013), we applied three gender (gen) features corresponding to the set {*hasMale, hasFemale, unknown*}. (Abdul-Mageed et al., 2014) suggest that there is a relationship between politeness strategies and sentiment expression. And gender variation research in social media has found that expression of linguistic politeness (cf. (Brown and Levinson, 1987)) differs based on the gender of the user: (Herring, 1994) identified gender differences in expressions of linguistic politeness in ways that interact with sentiment expression. (Herring, 1994) maintains that women are more likely than men to observe positive politeness through, e.g., thanking, while men prefer 'candor' and assertion of opinion, even when it conflicts with other people's opinions; such behaviors might interact with the type of subjectivity data carries.

User ID: The user id (uid) labels are inspired by research on Arabic Twitter [citation removed for blind review] showing that a considerable share of tweets is produced by organizations such as news agencies as opposed to lay users. Hence, two features from the set {*person, organization*} are employed for classification. The assumption we make is that tweets from persons will have a higher correlation with expression of subjectivity than those from organizations.

URL and Quotation: *(a). hasURL:* A binary feature indicating the existence of a URL in a tweet or lack thereof. *(b). hasQuotation:* A binary feature indicating whether a unit of analysis has quotation marks or not.

Existence of Latin: *hasLatin:* A binary feature indicating the existence of a Latin-alphabeted word in a tweet or lack thereof.

Speech-like Features: *(a). hasLetterRepetition:* A binary feature indicating the existence of a sequence of the same letter within a given word with a frequency > 3 in a tweet or lack thereof. *(b). hasLaughter:* A binary feature indicating the existence of the laughter word 'haha' or the laughter word 'hehe' in a tweet or lack thereof.

Emoticons: *(a). hasEmoticon:* A binary feature indicating the existence of an emoticon from the set { *';)', ':)', ':(', ';(', ':d', '(:', '):', ':D'*} in a tweet or lack thereof. *(b). hasPositiveEmoticon:* A binary feature indicating the existence of an emoticon from the set {*':)', ':d', ':; '(:', ':D'*} in a tweet or lack thereof. *(c). hasNegativeEmoticon:* A binary feature indicating the existence of an emoticon or emoticon-like interjection from the set {*':(', '):', 'ugh'*} in a unit of analysis or lack thereof.

Hashtags and Retweets: *(a). hasHashtag:* A binary feature indicating the existence of a hashtag '#' in a data point or lack thereof. *(b). hasMultipleHashtags:* A binary feature indicating the existence of two or more hashtags in a tweet or lack thereof. *(c). hasLongHashtag:* A binary feature indicating the existence of a hashtag of either length > 9 characters or with an underscore '_' in a data point or lack thereof. *(d). isRetweet:* A binary feature indicating whether a post is a retweet (has the prefix 'RE,' as is the norm in Twitter usage) or not.

Addressees: *(a). hasAddressee:* A binary feature indicating the existence of a username (as detected by the existence of an initial '@' sign in a string) in a tweet or lack thereof. *(b). hasMultipleAddressees:* A binary feature indicating the existence of two or more usernames in a tweet or lack thereof.

Punctuation: *(a). hasExclamation:* A binary feature indi-

[2] Although the focus of the current work is on the binary classification task of detecting whether a given tweet is OBJ or SUBJ, we also provide negative experiments on the sentiment data (as described in Section 5.).

Table 1: Data statistics

Data set	OBJ	S-POS	S-NEG	S-MIXED	# Tweets
MSA	960 (65.48%)	226 (15.42%)	186 (12.69%)	94 (6.41%)	1,466
DA	468 (30.21%)	257 (16.59%)	573 (36.99%)	251 (16.20%)	1,549
ALL	1,428 (47.36%)	483 (16.02%)	759 (25.17%)	345 (11.44%)	3,015

cating the existence of an exclamation mark in a data point or lack thereof. *(b). hasMultipleExclamation:* A binary feature indicating the existence of two or more exclamation marks in a tweet or lack thereof. *(c). hasQuestionMark:* A binary feature indicating the existence of a question mark in a unit of analysis or lack thereof. *(d). hasMultipleQuestionMark:* A binary feature indicating the existence of two or more question marks in a tweet or lack thereof.

Word Length: *(a). hasAvgShortWords:* A binary feature indicating whether the average word length of a unit of analysis is < 5 characters or not. *(b). hasAvgMediumWords:* A binary feature indicating whether the average word length of a tweet is at least 5 characters but < 7 characters or not. *(c). hasAvgLongWords:* A binary feature indicating whether the average word length of a tweet is > 7 characters or not.

Unit Length: *(a). hasShortLength:* A binary feature indicating whether the length of a unit of analysis is < 4 words or not. *(b). hasMediumLength:* A binary feature indicating whether the length of a unit of analysis is at least 4 words but < 8 words or not. *(c). hasLongLength:* A binary feature indicating whether the length of a tweet is at least 8 words but < 14 words or not. *(d). hasVeryLongLength:* A binary feature indicating whether the length of a data point is > 13 words or not.

4.2. Experimental Setup

Data Splits & Settings: We split the data into 80% training (TRAIN), 10% development (DEV), and 10% testing (TEST). We use three experimental settings as follows:

Individual Features (IVF): We add each of the individual features independently to the baseline bag-of-word (bow) setting and perform classification, thus measuring the utility of each of these features as combined with the simple bow baseline.

Whole Feature Set (WH): The whole feature set of 30 features is added to the baseline bow setting, and classification is performed. The way this setting is engineered is that any of the features that exist in any of the sentences used is added to the sentence vector, at the sentence level. This method allows identifying the utility of adding all the features combined on the classification task.

Feature Selection: Since some of the features may be more relevant than others to the task and since a feature can possibly perform differently based on the group of features it is used with, we also perform feature selection with a number of configurations, as follows:

Exhaustive feature group selection (FG): A search with all possible combinations of feature groups of the feature set is performed. In this setting, each group of the feature groups we described above is combined with zero or more groups, such that all possible combinations of the feature groups are considered. This method is better than the popular 'hill climbing' methods, whether in a forward or backward selection fashion. In *forward selection*, a given feature is added to a basic feature set, and if found useful, the feature is added to the basic feature set. Otherwise it is discarded, and the rest of the features are added in the same way to the basic feature set (which, after each iteration, includes more of the features of interest). The process continues until all features are considered, then the final performance is reported. Forward selection is described as 'hill climbing' search since it proceeds based on the potential gain each considered feature achieves in the classification process.

Backward selection is similar to forward selection, except that the classification starts with all the basic features, as well as all the features of interest, and a feature is dropped during each iteration to identify whether this ablation helps or hurts classification. The feature of interest is removed if its removal helps the classification, and the process is repeated. Like forward selection, backward selection proceeds based on potential gains removal of individual features can achieve. Exhaustive feature group search (FG) is better than hill climbing on feature groups in that during it, all possible combinations of groups of features are considered; hence any gains possible by any of such combinations are identified. This is different from hill climbing on feature groups, since hill climbing is not exhaustive and hence can miss possible feature group combinations that can achieve optimal performance. The down side of exhaustive search is its computational cost. However, this disadvantage is minor, since the process is performed offline. In addition, exhaustive search is practically possible only on a small feature set as the feature groups comprise here.

Monte Carlo feature selection (MC): A random sample of varying sizes from 1 to 30 of the individual features is added to the baseline bag-of-words setting, and classification is performed. This procedure is repeated $10K$ times, each time with a different random sample of a different size, such that different combinations of the individual features are possible. The Monte Carlo method is useful since, with a large number of iterations as in the case of $10K$, it is very likely that all possible combinations of individual features will be considered. The Monte Carlo method is preferred for mimicking exhaustive feature search with the individual 30 features. Attempting to perform individual feature exhaustive search with a procedure other than the Monte Carlo method would be extremely computationally costly and probably not needed, since processing the 30 social context features would mean 30! operations.

Procedure: We typically train classifiers on TRAIN, tune performance on DEV (e.g., to identify the performance of different sets of feature combinations and select overall best-performing feature set), and blind-test on TEST. For all the experiments, we use an SVM classifier with a linear kernel. We provide results on both DEV and TEST, as

appropriate.

Evaluation: Results are reported in terms of overall accuracy (acc) and F_1-score for the OBJ class (F_1-O) and the SUBJ class (F_1-S). Since the majority class in our training data is low ($= 52.64\%$), we use a baseline that is 10% higher. More specifically, we use performance with bag-of-words input (bow) on DEV (acc $= 62.67\%$) as our baseline.

4.3. Results

As Table 2 shows, on DEV, the whole feature set (WH) achieves a gain of 6.34% accuracy (acc) over the baseline bag-of-words (bow). In addition, the bow baseline is outperformed by the exhaustive group feature selection (FG) with 7.00% acc and by the Monte Carlo exhaustive feature selection method (MC) with 7.33% acc. Similarly, on TEST, the baseline is outperformed by WH and FG (with 12.74% acc for both cases), and by MC with 12.08% acc. All the gains are statistically significant ($p < 0.001$). Observably, TEST seems an easier set than DEV as indicated by its bow performance (at 66.23 acc) (vs. the baseline DEV bow, with acc $= 62.67$). Compared to the TEST bow performance, the models across all the experimental conditions on TEST are also highly successful and remain within statistical significance ($p < 0.001$): WH gains 9.18% acc, FG gains 9.18% acc, and MC gains 8.52% acc. We now turn to analyzing performance with each of our experimental settings.

FG Method: The FG method helps achieve the improvement with a number of feature group combinations. A consideration of these combinations shows that almost all the groups were chosen in one or another of them. In some of the selected combinations, some of the groups that were useful in other combinations were absent. For example, one of the combinations includes all the feature groups except the *hasLatin* feature, the *speech-like* features, and the *hashtag* features. These three specific feature groups were useful for the classification in other combinations that were also found to render the same classification improvement. This suggests that the FG feature selection method found intricate interactions among the groups. The importance of these groups of features is also reflected in the fact that the individual features within these groups were also selected via the MC method, whose performance we now turn to explaining.

MC Method: Similarly, in the MC method, several feature combinations were chosen. Again, an examination of these combinations shows that almost all the individual features were selected in one or another of the different combinations. For example, one of the combinations that achieved the best performance reported includes all the features except the three features *hasHashtag, hasQuestionMark,* and *hasIsMediumLength*.

IVF Method: Regarding experiments with the IVF feature engineering method, results show that the *gender (gen)* feature group, the *user ID (uid)* feature group, the *hasHashtag* feature, and the *hasURL* feature were useful for classification when added independently, as shown in Table 3.

Gender: The gender-based features proved useful for classification. In TRAIN, the distribution difference especially between the *female* and the *unknown* features within the objective and subjective classes is large enough to help classification: The *female* feature occurs in 25.14% of the subjective class data and 16.09% in the objective class data. For *unknown*, it occurs in 33.53% of the objective class cases and 14.92% of the subjective class. The *uid* group was also especially useful, with noticeably different distribution in TRAIN: The *person* feature occurs in 95.47% of the subjective class and in 80.29% of the objective class, whereas *organization* occurs in 19.71% in the objective class and 4.53% in the subjective class. A consideration of both TRAIN and DEV data shows that organizations seem to be more concerned with tweeting information objectively, perhaps as a way to gain credibility. After all, many of these organizations are news outlets interested in keeping their audiences' interest and trust, and (at least ostensibly,) unbiased coverage is important for them (Abdul-Mageed and Herring, 2008).

URL: The *hasURL* feature was also useful for classification. Based on TRAIN, tweets containing URLs are twice as likely to be in the objective (52.91%) class than the subjective class (23.17%). This is the case because URLs are more likely to be associated with information provision in the context of advertising, e.g., where users are encouraged to visit a website promoting some commodity. The following are two examples:

- (1) مهندس مدنى وارغب فى السفر للعمل فى السعودية http://bit.ly/iklvJh.

  Buck. 'mhnds mdnY wArgb fY Alsfr llEml fY AlsEwdyp http://bit.ly/iklvJh'

  Eng. '[I'm a] civil engineer and need a job in KSA http://bit.ly/iklvJh.'

- (2) برنامج يرنامج مشغلات المالتيميديا ٦ فى أخر إصداراته ٣٣١٠٥٠٢٠٣ http://goo.gl/fb/mBc2b.

  Buck. 'brnAmj yrnAmj m$glAt AlmAltymydyA 3.2.5.1306 fY Oxr ISdArAt http://goo.gl/fb/mBc2b.'

  Eng. 'Software software [sic] for playing multimedia 3.2.5.1306 in its latest release http://goo.gl/fb/mBc2b.'

Questions: Similarly, based on TRAIN, questions are more likely to occur in objective, information seeking tweets (7.83%) than in subjective tweets (6.84%). The following is an example of an objective question:

- (3) ممكن اعرف مين فى حركة ٦ ابريل من عين شمس عشان محتاج اتعاون معاهم.

  (**Buck.** 'mmkn AErf myn fY Hrkp 6 Abryl mn Eyn msEAn mHtAj AtEAwn mEAhm.'; **Eng.** 'Can I know who in April 6 Movement is in Ein Shams so that I contact them[?]').

Exclamation Marks: Unlike question marks, exclamation marks are quite expectedly more frequent in subjective cases than in objective cases in TRAIN. The *hasExclamation* feature occurred with a frequency of 6.84% in the subjective class and 3.12% in the objective class. Likewise,

Table 2: Results with whole set (WH), exhaustive group selection (FG), and Monte Carlo selection (MC)

				OBJ			SUBJ		
	setting	**acc**	**avg-f**	**prec**	**rec**	**f**	**prec**	**rec**	**f**
DEV	**base (bow)**	62.67	62.56	51.19	74.14	60.56	77.27	55.43	64.56
	WH	**69.00**	68.08	58.65	67.24	62.65	77.25	70.11	73.50
	FG	**69.67**	68.91	59.12	69.83	64.03	78.53	69.57	73.78
	MC	**70.00**	69.28	59.42	70.69	64.57	79.01	69.57	73.99
TEST	**bow**	66.23	65.69	54.67	70.09	61.42	77.42	63.83	69.97
	WH	**75.41**	73.96	68.10	67.52	67.81	79.89	80.32	80.11
	FG	**75.41**	73.96	68.10	67.52	67.81	79.89	80.32	80.11
	MC	**74.75**	73.27	67.24	66.67	66.95	79.37	79.79	79.58

Table 3: Individual features acquiring classification gains

					OBJ			SUBJ		
			Acc	**Avg-F**	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**
DEV		**bow**	62.67	62.56	51.19	74.14	60.56	77.27	55.43	64.56
	gen	**+**	63.67	63.37	52.23	70.69	60.07	76.22	59.24	66.67
	uid	**+**	65.00	64.72	53.5	72.41	61.54	77.62	60.33	67.89
	hasHashtag	**+**	63.33	63.2	51.81	74.14	60.99	77.61	56.52	65.41
	hasPositiveEmot	**+**	63.00	62.88	51.5	74.14	60.78	77.44	55.98	64.98
	hasURL	**+**	68.00	67.22	57.25	68.1	62.2	77.16	67.93	72.25
TEST	**gen**	**+**	66.89	66.12	55.63	67.52	61.00	76.69	66.49	71.23
	uid	**+**	68.20	66.91	57.81	63.25	60.41	75.71	71.28	73.42
	hasHashtag	**+**	66.89	66.36	55.33	70.94	62.17	78.06	64.36	70.55
	hasPositiveEmot	**+**	66.23	65.69	54.67	70.09	61.42	77.42	63.83	69.97
	hasURL	**+**	71.15	69.59	62.18	63.25	62.71	76.88	76.06	76.47

the *hasExclamationRepetition* feature was more frequent in the subjective class (with 1.57%) than in the objective class (0.67%). The following is an example of a subjective tweet employing multiple exclamation marks:

- (4) أنا بتفرج على فيديو دلوقتى هيجيبلى كوابيس !!! واضح إن فى منافس لأحمد زبايدر !!! مش قادر.

  Buck. 'OnA btfrj ElY fydyw dlwqtY hyjyblY kwAbys !!! wADH In fY mnAfs lOHmd zbAydr !!! m$ qAdr.'

  Eng. 'I'm watching a video right now[.] I'll have nightmares!!! Clearly, there's a competitor to Ahmad Spider!!! I can't take it.'

Emoticons: Although emoticons are usually viewed as symbols associated exclusively to subjective language, our annotators indeed assigned OBJ tags to a number of cases where positive emoticons occur. Positive emoticons, however, were more frequent in the subjective class (2.97%) than in objective class (1.18%). The following is an example of a smiley face in a subjective tweet:

- (5) شكرا وإذا فى أى ملاحظة ترى أنا إتقبل النقد .. :)

  Buck. '$krA wI*A fY OY mlAHZp trY OnA Itqbl Alnqd ..'

  Eng. 'Thanks[!] And let me know if you have any feedback[;] I take criticism.. :)'

Hashtags: The *hasHashtag* feature is also useful for classification as, in TRAIN, the feature is more frequent in the subjective class (with 0.49%) than in the objective class (with 0.34%). In DEV, the feature only occurred in subjective cases. Since hashtags are sometimes used to mark the topic of a tweet and have the potential to contribute to the popularity of a (trending) topic, they are used for campaigning in Twitter. Indeed, hashtags have played an important role in online activism in the Arab world (and elsewhere). In TRAIN, it is clear that political campaigning is an important function for which hashtags are used.

Sometimes users employ hashtags of more than one word, where the words are simply concatenated (potentially separated by an underscore). Longer hashtags are especially more frequent in the TRAIN subjective class (with a frequency of 0.08%) than in objective class (where they are totally absent). The same bias occurs in DEV (where their frequency is 0.54% in the subjective class and zero in the objective class). Examples of such hashtags are #egygilrs, #mubarakregrets, #wheniwasakidithought, #newegypt, and #3eshnaooshifna (Eng. 'Look what is happening').[3]

[3]In May 2010, when the dataset was being collected, hashtags in Arabic Twitter were exclusively in English, as Twitter did not allow use of hashtags with Arabic words. As explained earlier, even if a user wanted to use a hashtag with an Arabic word, the word would not be clickable. As of the writing of this paper, Arabic Twitter users employ a mixture of Arabic and English hashtags.

5. Negative Experiments

In order to test the performance of the feature set we propose here on sentiment classification, we run experiments with all the three settings on the polarity task (positive vs. negative) using the part of our data labeled with S-POS and S-NEG tags. With the SVMs classifier, we find that the WH, FG, and MC models outperform the baseline (bow on DEV, acc $= 68.09\%$) on the DEV data with 71.63%, 73.05%, and 73.76% respectively, but not on TEST where performance of these models drops to the same accuracy of 65.67% with each of the three settings. We conclude that the structural and social context features we propose are better suited for learning subjective (but not polar) language, and so we do not proceed with further experiments with GRUs on the polarity task.

6. Recurrent Neural Networks Models

Our deep learning model is based on a gated neural network. We now further introduced this class of methods. For notation, we denote scalars with italic lowercase (e.g., *x*), vectors with bold lowercase (e.g.,**x**), and matrices with bold uppercase (e.g.,**W**).

Recurrent Neural Network A recurrent neural network (RNN) is a neural network architecture that, at each time step t, takes an input vector $\mathbf{x}_t \in \mathbb{R}^n$ and a hidden state vector $\mathbf{h}_{t-1} \in \mathbb{R}^m$ and produces the next hidden state $\mathbf{h}_t$ by applying the following recursive operation:

$$\mathbf{h}_t = f\left(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}\right) \tag{1}$$

Where the input to hidden matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$, the hidden to hidden matrix $\mathbf{U} \in \mathbb{R}^{m \times m}$, and the bias vector $\mathbf{b} \in \mathbb{R}^m$ are parameters of an affine transformation and f is an element-wise nonlinearity. In theory, this design should enable an RNN to capture all historical information up to time step $\mathbf{h}_t$. In practice, however,RNNs suffer from the problems of vanishing/exploding gradients (Bengio et al., 1994; Pascanu et al., 2013) while trying to learn long-range dependencies.

LSTM Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) are designed to addresse this very problem of learning long-term dependencies: LSTMs are basically a variation of RNNs that are augmented with a memory cell $\mathbf{c}_t \in \mathbb{R}^n$ at each time step. As such, in addition to the input vector $\mathbf{x}_t$, the hidden vector $\mathbf{h}_{t-1}$, an LSTM takes a cell state vector $\mathbf{c}_{t-1}$ and produces $\mathbf{h}_t$ and $\mathbf{c}_t$ via the calculations below:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}^i x_t + \mathbf{U}^i \mathbf{h}_{t-1} + \mathbf{b}^i\right) \\
\mathbf{f}_t &= \sigma\left(\mathbf{W}^f x_t + \mathbf{U}^f \mathbf{h}_{t-1} + \mathbf{b}^f\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}^o x_t + \mathbf{U}^o \mathbf{h}_{t-1} + \mathbf{b}^o\right) \\
\mathbf{g}_t &= \tanh\left(\mathbf{W}^g x_t + \mathbf{U}^g \mathbf{h}_{t-1} + \mathbf{b}^g\right) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{2}$$

Where $\sigma(\cdot)$ and $\tanh(\cdot)$ are the element-wise sigmoid and hyperbolic tangent functions, $\odot$ the element-wise multiplication operator, and $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ are the *input, forget*, and *output* gates. The $\mathbf{g}_t$ is a new memory cell vector with candidates that could be added to the state. The LSTM parameters $\mathbf{W}_j$, $\mathbf{U}_j$, and $\mathbf{b}_j$ are for $j \in \{i, f, o, g\}$.

GRUs (Cho et al., 2014; Chung et al., 2015) propose a variation of LSTM with a *reset gate* $\mathbf{r}_t$, an update state $\mathbf{z}_t$, and a new simpler hidden unit $\mathbf{h}_t$, as follows:

$$\begin{aligned}
\mathbf{r}_t &= \sigma\left(\mathbf{W}^r x_t + \mathbf{U}^r \mathbf{h}_{t-1} + \mathbf{b}^r\right) \\
\mathbf{z}_t &= \sigma\left(\mathbf{W}^z x_t + \mathbf{U}^z \mathbf{h}_{t-1} + \mathbf{b}^z\right) \\
\tilde{\mathbf{h}}_t &= \tanh\left(\mathbf{W} x_t + \mathbf{r}_t * \mathbf{U}^{\tilde{h}} \mathbf{h}_{t-1} + \mathbf{b}^{\tilde{h}}\right) \\
\mathbf{h}_t &= \mathbf{z}_t * \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) * \tilde{\mathbf{h}}_t
\end{aligned} \tag{3}$$

The GRU parameters $\mathbf{W}_j$, $\mathbf{U}_j$, and $\mathbf{b}_j$ are for $j \in \{r, z, \tilde{h}\}$. In GRUs, the hidden state is forced to ignore a previous hidden state when the reset gate is close to 0, thus enabling the network to forget or drop irrelevant information. In addition, similar to an LSTM memory cell, the update gate controls how much information carries over from a previous hidden state to the current hidden state. GRUs are simpler and faster than LSTM, and so we use them instead of LSTMs in this work.

Network Architecture & Hyper-Parameters For GRUs, we use the same data split as described above with SVMs: 80% TRAIN, 10% DEV, and 10% TEST. We optimize the GRU hyper-parameters on the DEV set. We use a vocabulary size of $100K$ words, a word embedding vector of size 300 dimensions that we learn directly from the TRAIN, an input maximum length of 30 words, 2 epochs, and the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 0.001. We use a GRU layer with 500 units input, followed by 3 dense layers each with $1,000$ units. To regularize the network, we use dropout (Hinton et al., 2012) with a dropout rate of 0.5 after the first dense layer. For our loss function, we use binary cross-entropy. We use a mini-batch (Cotter et al., 2011) size of 128.

6.1. Results

Table 4 shows the best results acquired with feature engineering using our SVM classifier on both DEV and TEST from the previous section. As Table 4 shows, our GRUs model achieve an accuracy of 77.66% on DEV. This is $\sim 15\%$ higher than our baseline (`base`). On TEST, the model achieves 77.19, which is 14.52% higher than the baseline. This gain on TEST is also 10.96% higher than an SVM bag-of-words (`bow`) classifier on the same TEST set. Compared to the best accuracy on TEST with SVMs (acquired both with WH and FG, both at 75.41%), not to our surprise GRUs are 1.78% higher. This, however, emphasizes the utility of our feature set with the SVMs approach. Interestingly, the SVM models are better when it comes to detecting the `SUBJ` class: On TEST, our best SVMs models are a whooping 41.51% F_1-score higher than GRUs. The same observation holds with the results on DEV as well, with $\sim 21\%$ edge for the SVM classifier. It can be immediately seen that improvements are possible by simply combining predictions from the models with both approaches in an ensemble set up. We cast further investigation in this direction as potentially promising future research.

Table 4: Results with Gated Recurrent Neural Networks

				OBJ			SUBJ		
	setting	**acc**	**avg-f**	**prec**	**rec**	**f**	**prec**	**rec**	**f**
	base (svm bow)	62.67	62.56	51.19	74.14	60.56	77.27	55.43	64.56
DEV	**MC**	70.00	69.28	59.42	70.69	64.57	79.01	69.57	73.99
	GRU	**77.66**	76.54	81.59	89.45	85.34	62.30	46.34	53.15
TEST	**bow (svm)**	66.23	65.69	54.67	70.09	61.42	77.42	63.83	69.97
	WH	75.41	73.96	68.10	67.52	67.81	79.89	80.32	80.11
	FG	75.41	73.96	68.10	67.52	67.81	79.89	80.32	80.11
	GRU	**77.19**	76.02	77.90	95.98	86.00	70.97	26.51	38.60

7. Conclusion

We described successful models for learning subjective language from the Twitter domain. For learning, we introduced a framework of structural and social context features and showed its utility in classification with an SVMs approach. More specifically, our rich feature set totals 30 individual features that we also organize thematically into 11 different groups. Further, we introduced two feature selection methods, a Monte Carlo (MC) method for picking the best combinations of individual features and another method for exhaustive feature group selection (FG). We also analyzed the performance of the different combinations of feature groups as well as the individual successful features on the task, with illustrative examples. Our best performing model with these hand-crafted features on the blind test set is $> 12\%$ higher than our baseline. In addition, we carefully developed a highly successful deep gated recurrent neural network classifier that yields $\sim 14.50\%$ accuracy gains over our baseline. Comparing the classical SVMs classifiers to the GRUs on the task, we show the utility of our rich feature set and identify a promising route for future research where these approaches can be combined. Other future directions include expanding our work to other domains and possibly other languages.

8. Acknowledgements

This work was partially funded by UBC Hampton Grant #12R74395 and UBC Work Learn Award to the author. The research was also enabled in part by support provided by WestGrid (`https://www.westgrid.ca/`) and Compute Canada (`www.computecanada.ca`).

9. Bibliographical References

Abdul-Mageed, M. and Herring, S. (2008). Arabic and english news coverage on aljazeera.net. In *Proceedings of Cultural Attitudes Towards Technology and Communication 2008 (CATaC'08), Nimes, France*.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Abdul-Mageed, M., AlHuzli, H., and DuaaAbu Elhija, M. D. (2016). Dina: A multi-dialect dataset for arabic emotion analysis. In *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, page 29.

Abdul-Mageed, M., Alhuzali, H., and Elaraby, M. (2018). You tweet what you speak: A city-level dataset of arabic dialects. In *LREC*.

Abdul-Mageed, M. (2017a). Modeling subjectivity and sentiment in lexical space. In *Submitted*.

Abdul-Mageed, M. (2017b). Not all segments are created equal: Syntactically motivated sentiment analysis in lexical space. *WANLP 2017 (co-located with EACL 2017)*, page 147.

Aly, M. A. and Atiya, A. F. (2013). Labr: A large scale arabic book reviews dataset. In *ACL (2)*, pages 494–498.

Androutsopoulos, J. and Beißwenger, M. (2008). Introduction: Data and methods in computer-mediated discourse analysis. *Language@ Internet*, 5(2):1–7.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Seventh conference on International Language Resources and Evaluation, Malta. Retrieved May*, volume 25, page 2010.

Banfield, A. (1982). *Unspeakable Sentences: Narration and Representation in the Language of Fiction*. Routledge & Kegan Paul, Boston.

Benamara, F., Cesarano, C., Picariello, A., Reforgiato, D., and Subrahmanian, V. (2007). Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

Benamara, F., Chardon, B., Mathieu, Y. Y., Popescu, V., et al. (2011). Towards context-based subjectivity analysis. In *IJCNLP*, pages 1180–1188.

Benamara, F., Taboada, M., and Mathieu, Y. (2017). Evaluative language beyond bags of words: Linguistic insights and computational applications. *Computational Linguistics*.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bieswanger, M. (2013). 19. micro-linguistic structural features of computer-mediated communication. *Pragmatics of computer-mediated communication*, 9:463.

Blunsom, P., Grefenstette, E., and Kalchbrenner, N. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting*

of the Association for Computational Linguistics. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.

Breck, E., Choi, Y., and Cardie, C. (2007). Identifying expressions of opinion in context. In *IJCAI*, volume 7, pages 2683–2688.

Brown, P. and Levinson, S. (1987). *Politeness: Some universals in language usage*, volume 4. Cambridge Univ Pr.

Burger, J. D., Henderson, J., Kim, G., and Zarrella, G. (2011). Discriminating gender on twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1301–1309. Association for Computational Linguistics.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gülçehre, C., Cho, K., and Bengio, Y. (2015). Gated feedback recurrent neural networks. In *ICML*, pages 2067–2075.

Cohen, W. W. (1996). Learning trees and rules with set-valued features. In *AAAI/IAAI, Vol. 1*, pages 709–716.

Cotter, A., Shamir, O., Srebro, N., and Sridharan, K. (2011). Better mini-batch algorithms via accelerated gradient methods. In *Advances in neural information processing systems*, pages 1647–1655.

Cui, H., Mittal, V., and Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *AAAI*, volume 6, pages 1265–1270.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(3):131–156.

Deng, H., Han, J., Li, H., Ji, H., Wang, H., and Lu, Y. (2014). Exploring and inferring user–user pseudo-friendship for sentiment analysis with heterogeneous networks. *Statistical Analysis and Data Mining*, 7(4):308–321.

ElSahar, H. and El-Beltagy, S. R. (2015). Building large arabic multi-domain resources for sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 23–34. Springer.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, page 841. Association for Computational Linguistics.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

Hasan, K. S. and Ng, V. (2013). Extra-linguistic constraints on stance recognition in ideological debates. In *ACL (2)*, pages 816–821.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, pages 174–181. Association for Computational Linguistics.

Hatzivassiloglou, V. and Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 299–305. Association for Computational Linguistics.

Herring, S., Stein, D., and Virtanen, T. (2013). *Pragmatics of computer-mediated communication*, volume 9. Walter de Gruyter.

Herring, S. (1994). Gender differences in computer-mediated communication: Bringing familiar baggage to the new frontier. *Retrieved April*, 29:2002.

Herring, S. C. (2007). A faceted classification scheme for computer-mediated discourse. *Language@ internet*, 4(1):1–37.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Irsoy, O. and Cardie, C. (2014). Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*, pages 2096–2104.

Joachims, T. (2002). Support vector machines. In *Learning to Classify Text Using Support Vector Machines*, pages 35–44. Springer.

Kennedy, A. and Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, 22(2):110–125.

Kessler, J. S. and Nicolov, N. (2009). Targeting sentiment expressions through supervised ranking of linguistic configurations. In *ICWSM*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klenner, M., Fahrni, A., and Petrakis, S. (2009). Polart: A robust tool for sentiment analysis. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, volume 4, pages 235–238.

Labutov, I. and Lipson, H. (2013). Re-embedding words. In *ACL (2)*, pages 489–493.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Li, J., Luong, M.-T., Jurafsky, D., and Hovy, E. (2015).

When are tree structures necessary for deep learning of representations? *arXiv preprint arXiv:1503.00185*.

Lin, W.-H. and Hauptmann, A. (2006). Are these documents written from different perspectives?: a test of different perspectives based on statistical distribution divergence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1057–1064. Association for Computational Linguistics.

Liu, P., Qiu, X., Chen, X., Wu, S., and Huang, X. (2015). Multi-timescale long short-term memory neural network for modelling sentences and documents. In *EMNLP*, pages 2326–2335. Citeseer.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Hlt-naacl*, volume 13, pages 746–751.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Mohammad, S. M. and Kiritchenko, S. (2015). Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Montoyo, A., MartíNez-Barco, P., and Balahur, A. (2012). Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments.

Mourad, A. and Darwish, K. (2013). Subjectivity and sentiment analysis of modern standard arabic and arabic microblogs. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 55–64.

Ng, V., Dasgupta, S., and Arifin, S. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618. Association for Computational Linguistics.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318.

Persing, I. and Ng, V. (2014). Vote prediction on comments in social polls. In *EMNLP*, pages 1127–1138.

Purver, M. and Battersby, S. (2012). Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. Association for Computational Linguistics.

Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010). Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM.

Ren, Y., Zhang, Y., Zhang, M., and Ji, D. (2016). Context-sensitive twitter sentiment classification using neural network. In *AAAI*, pages 215–221.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C., et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., and Li, P. (2011). User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM.

Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014a). Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In *COLING*, pages 172–182.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014b). Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565.

Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Thomas, M., Pang, B., and Lee, L. (2006). Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 327–335. Association for Computational Linguistics.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.

Volkova, S., Wilson, T., and Yarowsky, D. (2013). Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *EMNLP*, pages 1815–1827.

Volkova, S., Bachrach, Y., Armstrong, M., and Sharma, V. (2015). Inferring latent user properties from texts published in social media. In *AAAI*, pages 4296–4297.

Wang, W., Chen, L., Thirunarayan, K., and Sheth, A. P. (2012). Harnessing twitter "big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust*

(PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom), pages 587–592. IEEE.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational linguistics*, 30(3):277–308.

Wiebe, J. (2000). Learning subjective adjectives from corpora. In *Proc.17th National Conference on Artificial Intelligence (AAAI-2000)*, pages 735–741, Austin, Texas.

Wiegand, M., Balahur, A., Roth, B., Klakow, D., and Montoyo, A. (2010). A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 60–68. Association for Computational Linguistics.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Wilson, T., Wiebe, J., and Hwa, R. (2006). Recognizing strong and weak opinion clauses. *Computational Intelligence*, 22(2):73–99.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Icml*, volume 97, pages 412–420.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Zhang, M., Zhang, Y., and Vo, D.-T. (2016). Gated neural networks for targeted sentiment analysis. In *AAAI*, pages 3087–3093.

Zhuang, L., Jing, F., and Zhu, X.-Y. (2006). Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM.