

AMERICAN UNIVERSITY OF BEIRUT

RESOURCES AND ANALYTICS FOR
OPINION MINING AND RECOMMENDER
SYSTEMS, WITH APPLICATION TO
ARABIC

by
GILBERT BADARO

A dissertation
submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
to the Department of Electrical and Computer Engineering
of the Maroun Semaan Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
March 2020

AMERICAN UNIVERSITY OF BEIRUT

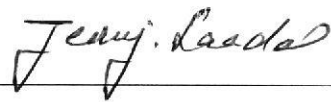
RESOURCES AND ANALYTICS FOR
OPINION MINING AND RECOMMENDER
SYSTEMS, WITH APPLICATION TO
ARABIC

by
GILBERT BADARO

Approved by:


Dr. Jean Saade, Professor
Electrical and Computer Engineering

Chairperson of Committee



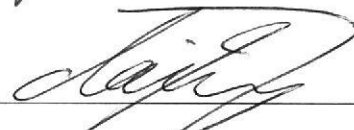
Dr. Hazem Hajj, Associate Professor
Electrical and Computer Engineering

Advisor



Dr. Zaher Dawy, Professor
Electrical and Computer Engineering

Member of Committee



Dr. Wassim El-Hajj, Associate Professor
Computer Science

Member of Committee



Dr. Nizar Habash, Associate Professor Member of Committee

Computer Science (New York University Abu Dhabi (NYU))

Jean' Loade

Dr. Mona Diab, Professor

Member of Committee

Computer Science (George Washington University (GWU))

Jean' Loade

Dr. Khaled Shaban, Associate Professor

Member of Committee

Computer Science and Engineering (Qatar University (QU))

Jean' Loade

Date of dissertation defense: December 18, 2019

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Badaro Gilbert
Last First Middle

Master's Thesis Master's Project Doctoral Dissertation

I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One** ___ year from the date of submission of my thesis, dissertation or project.
Two ___ years from the date of submission of my thesis, dissertation or project.
Three years from the date of submission of my thesis, dissertation or project.

Gilbert March 2, 2020
Signature Date

Acknowledgements

The PhD journey was full of ups and downs, but thanks to the support of friends, colleagues, professors, family and God, the downs were alleviated and became challenging experiences to learn from and improve.

I am so grateful that I have worked under the supervision of Prof. Hazem Hajj. I would like to thank him not only for his continuous support, help, knowledge, motivation and mentorship, but also for believing in my academic and research skills, and for encouraging me to pursue my PhD studies.

I would also like to express my gratitude to Prof. Nizar Habash for his valuable feedback, insights, ideas, knowledge and guidance. I have also been blessed to be a member of the Opinion Mining for Arabic (OMA) research team funded by Qatar National Research Fund (QNRF), and to get insightful remarks from Prof. Wassim El-Hajj and Prof. Khaled Shaban. I would also like to thank my dissertation committee members for their valuable inputs.

I would also like to thank CNRS-L/AUB for awarding me their prestigious scholarship in recognition of my high academic achievements. Special thanks for Prof. Brian Evans (University of Texas at Austin) and the Wireless Networking and Communications Group (WNCG) at UT Austin for hosting me as a research visitor during the academic year 2015-2016. It was also an honor to spend a year at CERN (2019-2020), CMS experiment, working with the Data Acquisition (DAQ) team and acting as the first ambassador of AUB in the CMS Induction Project for AUB, thanks to Dr. Frans Meijers (CERN), Dr. Remigius Mommsen (CERN), Mr. Martin Gastal (CERN) and the Dean's office of the Maroun Semaan Faculty of Engineering and Architecture (MSFEA) at AUB. I would like to thank QNRF, MSFEA, the Electrical and Computer Engineering (ECE) department and the University Student Faculty Committee (USFC) for their funding support to participate and to present my research work in top international conferences.

Last but not least, I would like to thank my family, my parents, Nada and Gaby, my sisters, Nathalie and Rosalie, my brother, Antoine, for standing by my side, and especially my mother, Nada, for her love, her support, her motivation and her unconditional efforts so we achieve the best. May the Lord protect them and bless them.

An Abstract of the Dissertation of

Gilbert Badaro for Doctor of Philosophy
Major: Electrical and Computer Engineering

Title: RESOURCES AND ANALYTICS FOR OPINION MINING AND
RECOMMENDER SYSTEMS, WITH APPLICATION TO ARABIC

This dissertation aims at exploring artificial intelligence solutions that help humans in their everyday life decisions such as where to stay, which doctors to consult, or which movie to watch. In particular, the dissertation goal is to address challenges and advance systems for recommender systems and opinion mining with special emphasis on Arabic. Opinion mining systems aim at extracting sentiment from data initiated by people such as data published in social networks. Recommender systems on the other hand use other peoples opinions and experiences to provide recommendations for users personal decisions. In this dissertation, several challenges are addressed to provide advances in resources and algorithms for recommender systems and opinion mining, in particular for Arabic. The dissertation includes a first of kind comprehensive survey for opinion mining in Arabic with a unique system and deployment perspective covering all relevant components including advances in NLP software tools, lexical and corpora resources, machine learning models, applications, and a roadmap for future development. Furthermore, to advance the field of automated opinion mining in Arabic, the first challenge was to develop lexical resources that are critical for the semantic interpretation of language. This problem is formulated as link prediction with the goal of linking large-scale Arabic lexical resources to previously developed English WordNet resources. Multiple natural language processing techniques are used including lemmatization, stemming, shallow and semantic similarity measures, feature extraction based on machine translation tables, and semantic word embeddings. For recommender systems, the challenges are in addressing sparsity of user-item

rating matrix for historical data, and the cold start problem for cases with no relevant history. To address these challenges, we propose two different models using collaborative filtering. The first approach is formulated as an optimization problem that aims at finding the best weights when combining the rating predictions of user-based and item-based collaborative filtering. The second approach consists of a multiresolution approach that computes affinity matrices between users and items at different granularity levels and projects the user-item rating matrix into a new multiresolution space. A selection of coefficients in the new space is used in order to reconstruct the user-item rating matrix in the original space and estimate the missing ratings. Finally, the dissertation includes detailed error analysis for the explored techniques. The developed resources have shown significant success and are heavily used in the research community. For instance, the Arabic Sentiment Lexicon, ArSenL, has received more than 95 citations. Success stories include its use for the construction of sentiment embeddings and the development of a light-weight phone-based sentiment classification system. The resources have been made publicly available.

Contents

Acknowledgements	v
Abstract	vi
1 Introduction	1
1.1 Dissertation Overall Objective	4
1.2 Dissertation Contributions	4
1.3 Dissertation Structure	5
2 Literature Review	6
2.1 Arabic NLP Tools	6
2.2 Arabic Lexical Resources	8
2.2.1 Arabic Sentiment Lexicons	8
2.2.2 Work on Emotion Lexicons	15
2.2.3 Work on WordNet Expansion	17
2.3 Arabic Sentiment Corpora	19
2.4 Opinion Mining Approaches	23
2.4.1 Lexicon-based Approaches	24
2.4.2 Feature Engineering “Supervised” Approaches	27
2.4.3 Deep Supervised Approaches	33
2.5 Applications	42
2.5.1 Opinion Mining Applications	42
2.5.2 Recommender Systems	45
3 Lexical Resource Expansion: Heuristic Techniques	50
3.1 Background on Existing Resources	50
3.1.1 Arabic WordNet (AWN)	50
3.1.2 English WordNet (EWN)	50
3.1.3 English SentiWordNet (ESWN)	51
3.1.4 Standard Arabic Morphological Analyzer (SAMA)	51
3.2 ArSenL: A Heuristic-based Link Prediction Approach	52
3.2.1 Methodology	52
3.2.2 Evaluation of ArSenL	55

3.3	EmoWordNet: Automatic Expansion of Emotion Lexicon Using English WordNet	59
3.3.1	Methodology	59
3.3.2	Evaluation of EmoWordNet	60
3.4	ArSEL: A Large Scale Arabic Sentiment and Emotion Lexicon . .	64
3.4.1	Methodology	64
3.4.2	Evaluation Using SemEval 2007 Dataset	66
3.4.3	Using SemEval 2018 Arabic Affective Tweets Dataset . . .	70
4	Lexical Resource Expansion: Machine–Learning Techniques	75
4.1	Developing Gold Dataset	77
4.1.1	Matching AWN Lemmas to SAMA Lemmas	77
4.1.2	Mapping AWN to EWN	80
4.1.3	Resulting Gold Set	80
4.2	Challenges in Shallow Link Prediction	80
4.3	Proposed Link Prediction Methods	85
4.3.1	Training and Testing Sets	85
4.3.2	Direct Matching	86
4.3.3	Using EWN Gloss and Extended Gloss	87
4.3.4	Using Machine Translation (MT) Tables	88
4.3.5	Using Word Embeddings	90
4.3.6	Fusion Model	91
4.4	Experimental Evaluation	92
4.4.1	Experimental Setup	92
4.4.2	Evaluation Approach	92
4.4.3	Evaluation of Link Prediction Techniques Based on Direct Matching	93
4.4.4	Evaluation of Link Prediction Techniques Based on EWN Gloss & Extended Gloss	102
4.4.5	Evaluation of Link Prediction Techniques Using MT Tables	102
4.4.6	Evaluation of Link Prediction Techniques Based on Word Embedding	107
4.4.7	Evaluation of Fusion Models	110
4.5	Proposed Final Expansion Model	112
4.6	Discussion and Comparison of Expanded Lexicon with Related Work	113
4.7	Summary	116
5	Applications	117
5.1	A Light Lexicon-based Mobile Application for Sentiment Mining of Arabic Tweets	117
5.1.1	Method Overview	117
5.1.2	Features and Mining Model	118
5.1.3	Mobile Application Development	119

5.1.4	Summary	122
5.2	Towards a Scalable Method for Accurate Recommender Systems Prediction	123
5.3	A Hybrid Approach for Recommender Systems	123
5.3.1	Choice of Weights α and β	124
5.3.2	Evaluation	126
5.3.3	Summary	127
5.4	Multiresolution Approach for Recommender Systems	127
5.4.1	Methodology	127
5.4.2	Evaluation	137
5.4.3	Summary	142
6	Conclusion and Guidelines for Future Work	143
A	Abbreviations	147
	Bibliography	149

List of Figures

1.1	An example of decision tasks users face on multiple occasions.	1
1.2	Simplified representation of the overall thesis objectives.	5
2.1	Collaborative Filtering process using user-based approach.	46
3.1	Mapping SAMA and AWN to ESWN.	53
3.2	Steps to map SAMA lemma to ESWN synset.	54
3.3	Overview of DepecheMood expansion approach.	59
3.4	Overview of ArSEL construction methodology.	65
3.5	Overview of ArSEL evaluation steps.	69
4.1	Overview graph of the training data development for AWN expansion. The dotted arrows represent the mapping described in section 4.1.	77
4.2	Walking example for computing similarity score between SAMA lemmas and EWN synsets based on MT tables.	89
4.3	Distribution of error reasons when using direct matching for false negatives and false positives by POS tag. FN1: Different Choices of English Words in SAMA versus EWN to Represent the Same Meaning. FN2: Limited SAMA English Gloss Terms. FN3: Different Inflections Used to Represent the Same English Lemma. FN4: Not Satisfying Minimum Threshold. FN5: Multiword Terms. FN6: Mismatch between AWN Lemma and SAMA Lemma. FP1: Matching to Homonyms between SAMA English Gloss Terms and EWN Synset Terms. FP2: Lack of Comprehensive Coverage of Links in the Gold Set. FP3: Mismatch between AWN Lemma and SAMA Lemma.	96
4.4	Comparison of count of lemmas with zero links and count of lemmas with links to EWN synsets outside the gold dataset in the test set when using raw MT and direct matching with raw SAMA English gloss terms.	106

4.5	Comparison of count of lemmas with zero links and count of lemmas with links to EWN synsets outside the Gold dataset in the test set when using lemmatized MT and direct matching with lemmatized SAMA English gloss terms.	107
4.6	Comparison of count of lemmas with zero links and count of lemmas with links to EWN synsets outside the gold dataset in the Test set when using Word Embedding Maximum and direct matching with lemmatized SAMA English gloss terms.	109
4.7	Comparison of count of lemmas with zero links and count of lemmas with links to EWN synsets outside the Gold dataset in the test set when using Fusion with boosting and best of direct matching for each POS tag.	111
5.1	Efficient opinion mining model in Arabic for mobile use.	118
5.2	Three-way ensemble decision trees sentiment classifier.	119
5.3	3-tier architecture of the mobile application.	120
5.4	Snapshot of different user displays.	122
5.5	Simulation results for different values of α and β	125
5.6	Simulation results for user-based, item-based and our proposed hybrid-base collaborative filtering. The y-axis represents the MAE.	127
5.7	Multiresolution approach for matrix completion.	128
5.8	Sample partition tree with three levels ($L = 3$) for items showing the groups of items' folders as nodes at multiple resolutions. At the finest level ($l = 3$), each node corresponds to a column in the user-item matrix.	131
5.9	Example of using dual affinity from items partition tree to compute similarity for users.	134
5.10	Mean Absolute Error for user-based and item-based collaborative filtering and for Harmonic Analysis approach.	138
5.11	Portion of the partition tree on users of the 1M dataset.	139
5.12	System Performance (in minutes) for running the proposed approach in terms of k_n for the three cases of Movielens dataset: 100 K, 1M, and 10M.	140
5.13	Performance (in minutes) and MAE versus choice of nearest neighbors (k_n) parameter for the 10M MovieLens dataset.	141
5.14	RMSE measure for different approaches using Netflix dataset.	142
6.1	Future work highlighted in orange in alignment with achieved contributions.	144

List of Tables

2.1	Summary of Arabic Sentiment Lexicons Part 1.	13
2.2	Summary of Arabic Sentiment Lexicons Part 2.	14
2.3	Arabic Opinion Mining Models: addressed challenges and drawbacks.	38
2.4	Summary of Arabic Opinion Mining models (1/3).	39
2.5	Summary of Arabic Opinion Mining Models (continued 2/3).	40
2.6	Summary of Arabic Opinion Mining Models (continued 3/3).	41
3.1	Examples of AWN synsets.	51
3.2	Detailed example for an EWN synset.	51
3.3	Detailed example for an ESWN synset.	52
3.4	Example of an entry in SAMA.	52
3.5	Summary of modifications performed to AWN lemmas in order to match them to SAMA lemma LDC form.	53
3.6	Sizes of created sentiment lexica.	55
3.7	Results of extrinsic evaluation. Numbers that are highlighted reflect the best performances obtained by the lexicons, without considering the baseline.	56
3.8	Samples of ArSenL showing entries originating from ArSenL-Eng and ArSenL-AWN.	58
3.9	Pearson Correlation values between predicted and golden scores. . .	62
3.10	F1-Measure results for emotion classification.	62
3.11	Examples of correctly classified headlines.	63
3.12	Examples of misclassified headlines.	63
3.13	Lexicons coverage.	64
3.14	Sample of ArSEL Arabic lemmas with emotion scores.	67
3.15	News' Headlines' examples to show differences between Google translations and human translations.	67
3.16	Mapping between SemEval and ArSEL emotion labels.	68
3.17	Pearson correlation values.	70
3.18	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.	70
3.20	Examples of correctly classified News' headlines from SemEval 2007.	71
3.21	Examples of misclassified News' headlines from SemEval 2007. . .	71

3.19	Distribution of emotion labels across the tweets.	72
3.22	Pearson Correlation results on SemEval 2018 Arabic tweets dataset.	73
3.23	Classification F1-score results on SemEval 2018 Arabic tweets dataset.	73
3.24	Examples of correctly classified Arabic tweets from SemEval 2018.	74
3.25	Examples of misclassified Arabic tweets from SemEval 2018. . . .	74
4.1	Example of the output generated by ALMOR [1, 2] for a given Arabic lemma.	80
4.2	Example of the mapping across the different resources.	81
4.3	Count of lemmas mapped from AWN to SAMA by POS tag.	81
4.4	Count of gold lemmas and links per POS tag.	81
4.5	Overall performance (%) of the different explored variations for Direct Matching per POS tag on the test set. Tec = Technique, Pre = Precision, Rec = Recall and Thr= Threshold.	93
4.6	Average performance (%) per lemma of the different explored variations for Direct Matching per POS tag on the test set. Tec = Technique, Pre = Average Precision per lemma, Rec = Average Recall per lemma, F1 = Average F1 per lemma and Thr= Threshold.	94
4.7	Assessment of predictions on the test set for the Noun POS tag for the different variations of link prediction when using direct matching between SAMA gloss terms against EWN synset terms.	94
4.8	Assessment of predictions on the test set for the Verb POS tag for the different variations of link prediction when using direct matching between SAMA gloss Terms against EWN synset terms.	94
4.9	Assessment of predictions on the test set for the Adjective POS tag for the different variations of link prediction when using direct matching between SAMA gloss terms against EWN synset terms.	95
4.10	Overall performance (%) of the different explored variations for link prediction using EWN Gloss and Extended Gloss. Tec = Technique, Pre = Precision, Rec = Recall and Thr= Threshold.	103
4.11	Average performance (%) per lemma of the different explored variations for link prediction using EWN gloss and extended gloss. Tec = Technique, Pre = Average Precision per lemma, Rec = Average Recall per lemma, F1 = Average F1 and Thr= Threshold.	103
4.12	Assessment of predictions on the test set for the Noun POS tag for the different variations of link prediction of using SAMA gloss terms against EWN gloss & extended gloss.	103
4.13	Assessment of predictions on the test set for the Verb POS tag for the different variations of link prediction of using SAMA gloss terms against EWN gloss & extended gloss.	103

4.14	Assessment of predictions on the test set for the Adjective POS tag for the different variations of link prediction of using SAMA gloss terms against EWN gloss & extended gloss.	104
4.15	Overall performance (%) of the different explored variations for link prediction using MT tables. Tec = Technique, Pre = Precision, Rec = Recall and Thr= Threshold.	104
4.16	Average performance (%) per lemma of the different explored variations for link prediction using MT tables. Tec = Technique, Pre = Average Precision per lemma, Rec = Average Recall per lemma, F1 = Average F1 per lemma and Thr= Threshold.	105
4.17	Assessment of predictions on the test set for the Noun POS tag for the different variations of link prediction when using MT Tables against EWN synset terms.	105
4.18	Assessment of predictions on the test set for the Verb POS tag for the different variations of link prediction when using MT Tables against EWN synset terms.	105
4.19	Assessment of predictions on the test set for the Adjective POS tag for the different variations of link prediction when using MT Tables against EWN synset terms.	105
4.20	Overall performance (%) of the different explored variations for link prediction using Word Embeddings. Tec = Technique, Pre = Precision, Rec = Recall and Thr= Threshold.	108
4.21	Average performance (%) per lemma of the different explored variations for link prediction using Word Embeddings. Tech = Technique, Pre = Average Precision per lemma, Rec = Average Recall per lemma, F1 = Average F1 per lemma and Thr= Threshold.	108
4.22	Assessment of predictions on the test set for the Noun POS tag for the different variations of link prediction when using Word Embeddings between SAMA gloss terms and EWN synset terms.	108
4.23	Assessment of predictions on the test set for the Verb POS tag for the different variations of link prediction when using Word Embeddings between SAMA gloss terms and EWN synset terms.	109
4.24	Assessment of predictions on the test set for the Adjective POS tag for the different variations of link prediction when using Word Embeddings between SAMA gloss terms and EWN synset terms.	109
4.25	Overall performance of the best of the different explored variations for link prediction and when using fusion approaches. Pre = Precision, Rec = Recall and Thr= Threshold.	111
4.26	Comparison to ArSenL. Pre = Precision, Rec = Recall.	114
4.27	Results of subjectivity and sentiment classification.	115
5.1	Example of a processed tweet.	121

5.2	Accuracy percentages for each classifier and for the full system. . .	122
5.3	Values of α and β used for testing the proposed method.	125
5.4	MAE for 1M and 10M MovieLens datasets.	139
5.5	Comparison of time performance per iteration for the 1M MovieLens Dataset	141

Chapter 1

Introduction

Nowadays, people are always trying to find ways to help them in their daily decisions such as the ones shown in Fig. 1.1: where to stay, what to eat, where to spend the weekend, what book to read, which movie to watch, where to invest, which doctor to consult, etc. . . With the sharp increase in the number of possible options to select from, people are more confused and need to spend more time reading thousands of reviews and comparing ratings to make their decision. For example, there are around 4k movies in the US Netflix library¹ to choose from. Hence, there is a need for artificial intelligence solutions to help in the decision making process, specifically, to provide personalized recommendations using opinion mining systems and recommender systems applied on data from social media.

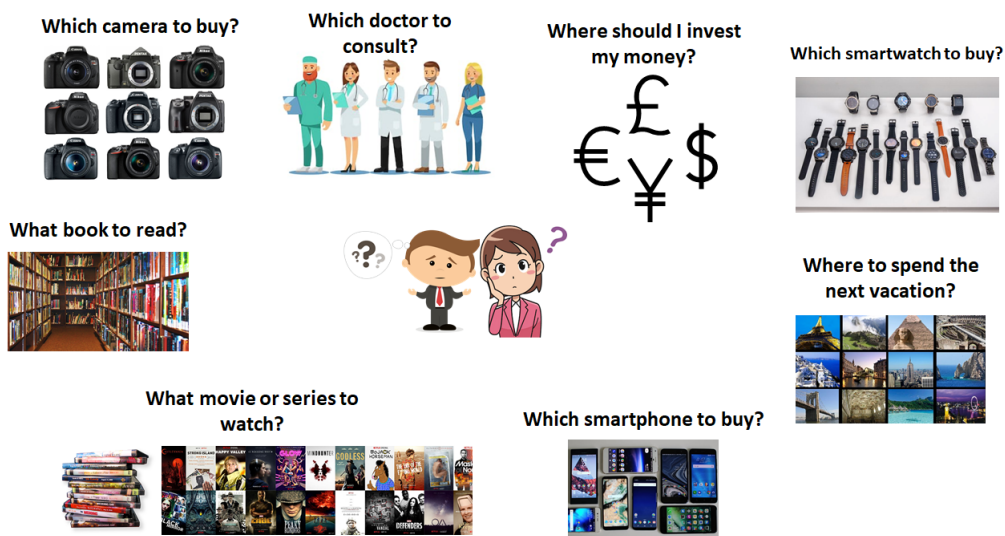


Figure 1.1: An example of decision tasks users face on multiple occasions.

¹<https://www.finder.com/netflix-movies>

Everyday more people are being part of the social media and start sharing their opinions, comments and pictures on social media websites such as Facebook, Twitter and Instagram. With the increase of social media websites and blogs, the users are no more only consumers of information but also producers of information [3]. The high amount of user generated data on social media is a great knowledge which was not easily available before the explosion of social media. People are often interested in determining other people's opinions when, among other examples, seeking to buy products, sensing the public opinion on certain issues, or identifying trends. Governments and politicians are often interested in opinion (or sentiment) mining for defining policies and campaign strategies. As a result, opinion mining, which aims at automatically extracting people's opinions, has found significant importance in Politics, Social Media, and Business. The task of sentiment classification in texts consist of detecting whether a given word, sentence or document has a positive, negative or neutral opinion. In fact, sentiment classification has been well studied for English language [4, 5, 6], and several lexical resources exist such as English WordNet [7] and English SentiWordNet [8, 9]. Recently, with the advance of computing technologies and availability of large amount of cheap processing memory, deep learning models have shown promise for sentiment classification and have outperformed previously existing state-of-the-art approaches [10, 11]. Furthermore, the application of Opinion Mining in Arabic text (OMA) is a timely subject, given the importance of the Arabic language, which emerged as the 5th most-spoken language worldwide and recently became a key source of Internet content and now stands as the 4th most used language on the Internet [12]. Arabic opinion mining has been an active area of research, but still has many open challenges.

Arabic language is both morphologically rich and highly ambiguous, and hence is challenging for most NLP applications. According to [13], a complete part-of-speech (POS) tagset in Modern Standard Arabic (MSA) has over 300K tags, whereas English has around 50. Also, MSA words have 12.2 morphological analyses, on average, whereas English has only 1.25 analyses per word. This high ambiguity is primarily due to Arabic orthography, which almost always omits the diacritics that are used to specify short vowels and consonant doubling. Furthermore, Arabic has complex morpho-syntactic agreement rules and a lot of irregular forms. Finally, although MSA is the official form of Arabic, it is no one's mother tongue. In fact, Arabic has a large number of dialectal variants that are as different from MSA as Romance languages are different from Latin. These different dialects are used for everyday communication but do not follow MSA language rules. As a result, it becomes challenging to create NLP resources for Arabic dialects [14, 15]. Moreover, dialectal Arabic includes several misspellings and typical Arabic NLP morphological tools do not perform well on dialects [16]. The lack of writing standards is particularly challenging for processing Arabic dialect big data such

as blogs and tweets, where we find a wide range of variations in spelling and even script choice, e.g., using Latin characters to write Arabic (aka, Arabizi).

Despite the availability of several resources and models for English sentiment classification, this is not the case for the Arabic language. In fact, the Arabic language through its morphological complexity and limited resources makes the development of sentiment lexical resources more challenging. It is known that the availability of sentiment lexicons improves significantly the accuracy of sentiment classification for texts. While there have been some efforts to develop Arabic sentiment lexicons, they still suffer from many deficiencies: limited size, unclear usability plan given Arabics rich morphology or non-availability publicly. Unlike English, there is still no publicly available large scale Arabic SentiWordNet [17, 18, 19]. Although one could argue that there exists an Arabic WordNet (AWN) [20] and one could apply a similar algorithm that was used to semi-automatically create English SentiWordNet (ESWN), AWN is of small size and contains a lot of inconsistencies in terms of lemma writing making it less useful for performing Arabic text processing. In order to create a large scale publicly available sentiment lexicon for Arabic in an automatic way, existing Arabic lexical resources are harvested: AWN and the Standard Arabic Morphological Analyzer (SAMA) [21] and English resources: EWN and ESWN. Moreover, developing resources and models for the Arabic language is important given the large number of Arabic users of the social media and the huge amount of Arabic opinionated text on the web. For instance, there were more than 195 million Facebook users in the Arab World by the end of 2017 [22]. Thus, there is a need to develop Arabic Sentiment lexicons and optimized sentiment models. These models can also be integrated into recommender systems allowing the development of improved user recommendation models with attention to Arabic users.

Similarly, with the advance of e-commerce activity and shopping with Web 2.0, recommender systems continue to attract further the attention of data scientists. A recommender system aims at providing user-specific content synthesized from an overwhelming large amount of information which may not be necessarily of interest to the user [23]. Moreover, recommender systems help people in several daily activities such as: what book to read, what movie to watch, what music to listen to, what restaurant to visit, what touristic travel packages to choose, etc. [24]. For instance, by filtering on books on Amazon.com, one finds that there are more than 60 thousand new releases in the last 30 days (December 31, 2018). 60 thousand books will require a lot of effort and time by the user to browse through whereas a recommender system provides the user a specific list of recommended books that match the user preferences. Recommender systems help in keeping the relevant items to their specific profile, i.e., a person x may see a different content than a person y . Recommender systems are also a helpful tool for businessmen or businesswomen for marketing and for customized advertisements [25]. Several approaches have

been proposed in the literature for recommender systems. Traditional methods rely on the ratings provided by the users. These methods can be classified into: collaborative techniques, content-based techniques, hybrid models, and preference-based methods [26, 27]. Some researchers decided to use social media to extract several user-context features and improve the recommendation by proposing context aware recommender systems [28]. The context data can be for example time, location, profession, company of other people, etc. More recent work integrated a larger amount of textual information by extracting the sentiment of the user for improved recommendation [29]. Several challenges of recommender systems still exist, and that we address including accuracy, sparsity of the user-item matrix, and cold start users.

1.1 Dissertation Overall Objective

The overall goal of the dissertation shown in Fig. 1.2 is to develop resources and models to suggest personalized content to the users to help in their daily decisions. The objective is achieved by analyzing the opinionated social media content which can be manifested in different forms such as opinionated text or ratings. To process the textual content, specifically the Arabic one, the objective is to develop sentiment and emotion lexical resources to enable opinion mining systems. To process the ratings, that are usually presented in a user-item rating matrix, models for recommender systems are developed. The output of the opinion mining system can also be used to translate the textual opinions into ratings that can be used as input to the recommender systems when explicit ratings are not available. The output of the decision making system is of great benefit not only for individual users but also for marketers, businesses, government and politicians.

1.2 Dissertation Contributions

The dissertation presents the following research contributions:

- Presenting a detailed comprehensive literature survey on different components essential for opinion mining systems including NLP tools, lexical resources, sentiment corpora, models and applications with insights on research roadmap.
- Developing automatically large scale Arabic sentiment and emotion lexical resources.
- Analyzing and evaluating multiple link prediction techniques for automatic Arabic WordNet improvement and expansion.

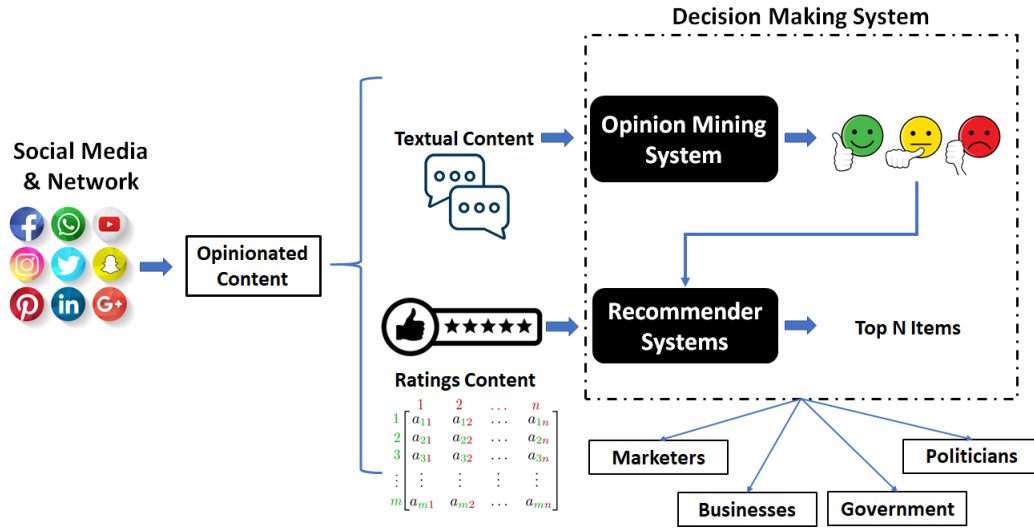


Figure 1.2: Simplified representation of the overall thesis objectives.

- Developing a benchmark dataset for automatically evaluating Arabic lexical resource linking to English WordNet.
- Enabling light weight efficient sentiment classification application for Arabic texts on mobile.
- Developing language and context independent accurate recommender systems models using a hybrid approach with collaborative filtering and using harmonic analysis.
- Publications of 2 ACM transactions [30, 31] and 14 conference papers [32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45], with more than 285 citations based on Google scholar citations.

1.3 Dissertation Structure

The dissertation is organized as follows. Chapter 2 presents a detailed literature review on Arabic opinion mining systems and recommender systems. Chapter 3 and chapter 4 present the details related to the automatic development of Arabic lexical resources using heuristic techniques and machine learning techniques respectively. Chapter 5 presents applications for opinion mining covering a prototype of a mobile application for opinion mining on Arabic tweets and scalable models for improving the accuracy of context and language independent recommender systems. Finally, a conclusion and guidelines for future work are presented in chapter 6.

Chapter 2

Literature Review

The objective of this literature survey is to give a general overview about some important concepts related to the dissertation work. We start by giving an overview on existing Arabic NLP tools followed by a detailed summary on existing Arabic sentiment and emotion lexical resources. Then we present a summary of efforts for expanding WordNets in different languages including Arabic. A brief overview on Arabic sentiment corpora is also provided followed by a summary on Arabic sentiment classification models. Last but not least, the chapter is concluded with an overview of applications including recommender systems' models. Most of the discussed work related to opinion mining is published in the journal article [30] entitled "A Survey of Opinion Mining in Arabic: A Comprehensive System Perspective Covering Challenges and Advances in Tools, Resources, Models, Applications, and Visualizations".

2.1 Arabic NLP Tools

Many tools and resources have been developed to perform Arabic NLP functionalities, with some reaching competitive performance levels compared to languages that are extensively studied such as English [13]. Central to these efforts was the creation of corpora that are annotated with linguistic information. Examples include the Penn Arabic Treebank (PATB) [46] that includes phrase structure trees along with POS tags and morphological information, and the Columbia Arabic Treebank (CATiB) [47] that contains dependency trees. Such corpora have been instrumental in the development of different Arabic NLP tools for tokenization [48, 49], diacritization [49, 50], POS tagging and syntactic parsing [51, 52] and morphological analysis [21].

It is worth mentioning that most Arabic NLP tools are designed and trained on MSA datasets. Some of these tools were developed to perform specific functionalities. For instance, segmentation is tackled in [53] by training deep recurrent neural models such as Long Short-term Memory (LSTM). Systems for

Named Entity Recognition (NER) [54, 55] and diacritization [56] were developed as well. On the other hand, some other tools are comprehensive in the sense that they perform several NLP tasks under one framework. An example of such tools is MADAMIRA [57] that performs morphological disambiguation, which encompasses tokenization, POS tagging, base phrase chunking (BPC), diacritization, lemmatization, gender, number, person, in addition to NER. MADAMIRA is based on the ancestor systems: MADA [49, 58] and AMIRA [59]. Also, in [60], a grammar analyzer is developed based on basic morphology analysis using the Buckwalter Arabic morphological analyzer (BAMA-v2.0) [61], stemming, POS, BPC and finally a rule based system applying 400 Arabic grammar rules. Other software tool examples include TectoMT [62], ASMA [63] and NERA [55].

Although dialectal Arabic (DA) is quite prevalent, especially in social media, it has received little attention with a few early efforts [64, 65]. With the development of accurate NLP tools for MSA, more attention has been given to DA. For instance, a small corpus of Egyptian Arabic was annotated for morphological segmentation to learn segmentation models [66]. Al-Sabbagh & Girgu in [67] described a supervised tagger for Egyptian Arabic social networking. Habash et al. in [68] presented the first large-coverage morphological analyzer for Egyptian Arabic. A tool for automatic identification of dialectal Arabic (AIDA) was also developed in [69]. Dialects of the Arabian Gulf region were also studied very recently in [70]. Given the inherited similarity between MSA and DA, there have been efforts to map from DA back to MSA, in order to exploit the existing MSA resources [56, 71, 72]. Additional work has also been done for dialect identification [73, 74, 75]. Recently, given the significant performance improvements achieved for sentiment analysis when using word vector representations, Soliman et al. in [76] developed AraVec. AraVec is a pre-trained distributed word representation open source project which aims to provide the Arabic NLP research community word embedding models. The first version of AraVec provides six different word embedding models built from three different Arabic content domains; Tweets, World Wide Web pages and Wikipedia Arabic articles. The total number of tokens used to build the models amounts to more than 3.3 billion.

In summary, several Arabic NLP tools and resources, specifically those targeting MSA, have been developed achieving performances that are comparable to those of English language. Further efforts are still needed to achieve better results when it comes to DA given the prevalence of using different dialects across the Arab World.

2.2 Arabic Lexical Resources

2.2.1 Arabic Sentiment Lexicons

The establishment of advanced Arabic NLP tools and resources, as discussed in subsection 2.1, is useful for opinion mining as syntactic and morphological features can mitigate the impact of Arabic’s complex morphology [77, 78, 79]. Also, it is crucial to equip opinion models with deeper insights into the text semantics. Sentiment lexica are crucial resources for most sentiment analysis algorithms [80]. Several researchers worked on developing Arabic sentiment lexica to improve the accuracy of OMA. In general, there are two approaches for creating such lexicons; manual and automatic. The manual approach consists of manually determining the sentiment of a list of Arabic words that are extracted from a certain domain or dataset. The resulting lexicon is usually highly accurate but limited in size due to the time-consuming task of annotation. Several resources were developed manually, from which we mention the following. ArabSenti [77] contains 3,982 adjectives extracted from 400 documents belonging to the ATB part 1 V3.0 [46]. These adjectives were manually labeled by two Arabic native speakers as positive, negative or neutral and were reviewed by a linguist expert. Similarly, SIFAAT [18] was manually developed and consisted of 3,325 adjectives labeled as positive, negative or neutral. Using SIFAAT showed significant improvement in the accuracy results of the proposed subjectivity and sentiment analysis system. Al-Kabi et al. in [81] collected 1,080 Arabic reviews that only include Arabic characters. They applied first a set of preprocessing steps: removal of digits, punctuations, symbols and special characters, normalization and tokenization. They annotated the reviews for sentiments (positive versus negative) as well as for domain (8 domains in total). Based on annotation, they created two general purpose lexicons one including positive terms and another one consisting of negative terms. Moreover, eight domain specific lexicons were manually developed. Al-Rowaily et al. in [82] presented BiSAL, a sentiment lexicon specifically related to cyber threats, radicalism and conflicts. BiSAL is composed of an English version that includes 279 sentiment terms and an Arabic version of 1,019 terms. The resource is publicly available.¹ Abdulla et al. in [83] also used a manual approach for creating their lexicon by translating 300 English seed words from SentiStrength [84] to Arabic. This seed set was then expanded using synonyms and antonyms, and by including emoticons. Unlike ArabSenti and SIFAAT, this lexicon does not include neutral terms and is not publicly available. NileULex is a recently developed sentiment lexicon [85, 86] that includes single word terms as well as compound phrases for MSA and dialects. Although the extraction of the terms and compound phrases

¹<http://www.abulaish.com/bisal>

was done automatically from social media, the annotation was performed manually. The authors chose single word terms to be as unambiguous as possible and compound phrases were used to overcome ambiguity in single word terms. NileULex consists of 5,953 expressions annotated either as positive or negative. NileULex was used by NileTMRG team [87] which participated at SemEval 2016 Task 7 [88]. El-Beltagy in [89] extended NileULex by presenting a method for automatically assigning strength scores or weights to NileULex entries as well as making the resulting lexicon “WeightedNileULex” publicly available. In [90], Ibrahim et al. presented AIPSeLEX, idioms/proverbs sentiment lexicon for MSA and DA. AIPSeLEX was manually collected and annotated at sentence level with semantic orientation (positive or negative). AIPSeLEX consists of 3,632 phrases annotated for sentiment with the help of three native Arabic speakers.

The larger the lexicon, the better the coverage of the language vocabulary, and thus the better the sentiment modeling. Therefore, automatic approaches to develop sentiment lexica are required. Although an automatic approach might be prone to errors and noise, it is cheap and less time-consuming. Usually, automatic approaches are supported with some manual efforts to allow for automatic expansion. They usually consist of harvesting existing sentiment lexical resources in English and trying to create equivalent versions in Arabic. For instance, Mourad & Darwish in [91] utilized the manually developed ArabSenti to perform automatic expansion through graph reinforcement. They translated ArabSenti into English and then used machine translation tables of English-MSA and English-DA to enrich ArabSenti with new MSA and DA terms. They also translated the MPQA lexicon [92] from English to Arabic using the Bing machine translation tool, and combined all lexica together to use them in their opinion mining classification system. However, the authors did not report the total number of terms they finally obtained. Alhazmi et al. [17] linked the Arabic WordNet (AWN) [20] to the English SentiWordNet (ESWN) [8, 9] through synset offset information. Their approach had limited coverage (around 10K lemmas only) and did not define a process for using the lexicon in practical applications given Arabic’s complex morphology. Furthermore, it was not made publicly available and was not evaluated in the context of a sentiment classification application. In [93], starting with an English sentiment lexicon derived from the General Inquirer English sentiment lexicon, Hassan et al. proposed an automatic approach for creating a lexicon in other languages by using semantic relationships available through WordNet. They applied their approach on Arabic and Hindi using English WordNet, Arabic WordNet and Hindu WordNet. Following a similar concept, Mahyoub et al. [94] used AWN 2.0 synset relations such as ‘near_antonym’ and ‘near_synonym’ to automatically expand a seed list translated from English to Arabic where the list is proposed by Turney and Littman [95]. Since they could not cover all words in the AWN with the initial seed list, they randomly picked

up terms and added them to the list. The resulting lexicon had a size of around 7.6K words and improved sentiment classification results when used in addition to other stylistic and syntactic features. Badaro et al. in [36] benefited from the success and availability of the ESWN, and developed a lemma-based large-scale Arabic sentiment lexicon: ArSenL. This lexicon is the result of combining two sub-lexicons. The first consists of mapping AWN 2.0 to ESWN 3.0 by going through sense map files across English WordNet versions similar to the work of Alhazmi et al. [17], but with an important addition: standardizing the lemma format of AWN terms to LDC format. This step is important in making the resource easy to integrate with other Arabic NLP tools. The second sub-lexicon, ArSenL-Eng was the result of mapping SAMA [21] directly to ESWN 3.0 by matching SAMA gloss terms to ESWN synset terms. ArSenL is publicly available and includes around 29K lemmas along with their corresponding POS tag, EWN synset ID and ESWN sentiment scores. It also improved accuracy in subjectivity and sentiment classification tasks. Recently, Badaro et al. in [45] presented ArSEL, a large scale Arabic Sentiment and Emotion Lexicon, an extension to ArSenL with the addition of 8 emotion scores to most of ArSenL Arabic lemmas. The emotion scores are extracted from a WordNet based English emotion lexicon, EmoWordNet [43]. Eskander and Rambow developed SLSA [96] using almost the same approach as ArSenL but they used instead of SAMA, AraMorph [97] which is the publicly available version of SAMA. Moreover, when matching gloss terms to synset terms, they applied different heuristics and back-off measures in order to ensure higher coverage but at the expense of additional noisy mappings. SLSA consists of around 35K Arabic lemmas and is publicly available. Similarly, Sabra et al. in [98] developed a sentiment lexicon for Arabic by first developing an English sentiment lexicon and then mapping it to Arabic. Starting with a seed list of 18 words equally split between positive and negative, the authors utilized WordNet semantic relations to expand the list. During the expansion, the depth needed to reach a certain term was recorded and then used to compute scores for positive and negative sentiments. The result was an annotated EWN for sentiment. Next, the authors mapped the EWN terms to SAMA lemmas through synset terms to SAMA gloss matching. The lexicon that consists of around 78K entries was tested on OCA corpus and achieved comparable performance to ArSenL with a simple sentiment classification model. Abdulla et al. [99] adopted a semi-automatic approach to develop their sentiment lexicon. They manually translated from English to Arabic 300 terms from SentiStrength and then they used synonym tables to expand the initial list. For the automatic part, they have translated the remaining of SentiStrength using Google translate and they also investigated using an annotated corpus for sentiment to extract positive and negative words using a Term-Frequency weighting scheme. Abdul-Mageed and Diab [100] extended their manually developed sentiment lexicon (SIFAAT) automatically by using machine translation and statistical formulation based on

point-wise mutual information to create SANA. SANA included 224,564 entries which cover Modern Standard Arabic (MSA) as well as Egyptian and Levantine dialects. These entries are not distinct and possess many duplicates. Developing SANA involved gloss matching across Arabic lexical resources such as THARWA [101] and SAMA [21] and English sentiment resources, the Affect Control Theory lexicon [102] and ESWN [9]. It is also composed of different lexicons such as Yahoo Maktoob, a dataset from Twitter and an automatically translated YouTube comments dataset [103]. Unlike SIFAAT, SANA was not tested in a sentiment classification task and is not publicly available. Chen and Skiena in [104] proposed an expansion approach of an English sentiment lexicon to more than 130 languages using knowledge graph construction. Using Wiktionary, Google Translate, transliteration and WordNet, the authors tried to get semantic links between words in their different translations. Using graph propagation algorithm and the semantic links, they extended sentiment polarities from a set of terms to their neighbors. For Arabic, they were able to construct a sentiment lexicon of 2,794 words starting with a set of 1,422 positive and 2,956 negative English terms. Al-Ayyoub et al. in [105] developed a sentiment lexicon of 120K stems that were collected by crawling the web specifically AlJazeera.net and by using Abuaiadah dataset for document classification [106]. ArSeLEX [107, 108] is an automatically generated and publicly available sentiment lexicon that contains 5,244 adjectives. ArSeLEX was developed by getting synonyms and antonyms of 400 adjectives manually annotated for sentiment. El Sahar & El-Beltagy in [109] used a set of lexico-syntactic rules to automatically extract Arabic phrases that represent opinion. They applied the rules on DA, specifically Egyptian Cairene dialect. Starting with a seed set manually translated from English to Arabic, the authors defined a set of patterns that could capture subjective text in Arabic. After extracting the subjective slangs, pointwise mutual information (PMI) was used to determine the polarity of the extracted phrase using an annotated set of Tweets. Out of 7.5M cleaned Tweets, they were able to extract 633 expressions with an 89% precision. El Sahar & El-Beltagy in [110] used a supervised learning model to generate a sentiment lexicon with positive and negative labels. From an annotated corpus of 35K sentences, they extracted unigrams and bigrams and used them for sentiment classification with SVM. The terms with highest positive coefficients were labeled as positive sentiment and the terms with lowest negative coefficients were considered as holding negative opinion. Arabic Senti-Lexicon [111] was developed using a combination of automatic approaches followed by manual adjustments. Terms from MPQA lexicon [92] were translated to Arabic using Google Translate and the translation was manually adjusted. The list was expanded by adding synonyms and antonyms. Two types of annotations were provided for the words: a manual annotation provided by three annotators consisting of a score on a scale varying between -5 to +5 with -5 being very negative and +5 very positive. A

second automatic score was computed using PMI such that the sum of the positive and negative scores is equal to 1. Arabic Senti-Lexicon consists of 13,760 terms and covers MSA as well as dialects. Al-Twairesh et al. in [112] focused on developing sentiment lexicons for DA in order to improve the accuracy of opinion mining systems applied on Twitter data. The authors first collected around 2.2 million tweets that included specific positive and negative seed words. These seed words in addition to emoticons helped in performing automatic sentiment annotation for the collected tweets. Using the annotated twitter dataset, Al-Twairesh et al. created two sentiment lexicons: AraSenti-Trans and AraSenti-PMI. For AraSenti-Trans, the authors ran the tweets through MADAMIRA [57], extracted Arabic lemmas and tried to match the provided English gloss with existing English sentiment lexicons: Liu Lexicon [113] and MPQA [92], in order to assign sentiment labels for the Arabic lemma based on defined matching rules. AraSenti-Trans included around 132K Arabic terms with around 60K positive and 72K negative. AraSenti-PMI is generated by computing PMI for all the words occurring at least more than 5 times in the positive and negative sets of tweets and then generating a sentiment score for the word using its PMI. AraSenti-PMI included around 57K positive terms and 37K negative terms. They evaluated both lexicons in sentiment classification tasks on different Twitter datasets: AraSenti-Tweet [114], ASTD [115] and RR [116], and obtained best average F1-scores of 88.92%, 59.8% and 63.60% on the three datasets respectively. The authors in [117] applied the annotation technique of Best-Worst Scaling to obtain real-valued sentiment association scores for words and phrases in three different domains: general English, English Twitter, and Arabic Twitter. They showed that on all three domains the ranking of words by sentiment remains remarkably consistent even when the annotation process was repeated with a different set of annotators. The authors also asked the annotators to indicate the minimum perceptible difference in sentiment strength between two terms. The annotated data is publicly available.² For annotation, they utilized CrowdFlower (now known as Figure Eight). Three different lexicons were created: (1) SemEval 2015 English Twitter Lexicon consisting of 1,515 high-frequency English single words and simple negated expressions commonly found in tweets. (2) SemEval 2016 Arabic Twitter Lexicon that includes 1,367 most frequent terms and expressions in Arabic tweets. (3) SemEval 2016 General English Sentiment Modifiers lexicon also known as Sentiment Composition Lexicon for Negators, Modals and Degree Adverbs: it consisted of 1,621 positive and negative single words in addition to 1,586 high-frequency phrases. Last but not least, Assiri et al. [118] created a sentiment lexicon targeting Saudi dialects by manually annotating terms extracted from Saudi social media and adding it to a modified version of ArSenL. The modified version of ArSenL was obtained by removing

²www.saifmohammad.com/WebPages/BestWorst.html

Lexicon	Lemmas	# Entries	Labels	Method	Tested for SA	Source	POS Tags	MISA	Dialect	Emoticons	Available
ArabSenti [77]	Not specified	3,982	Pos, Neg, Obj	Manual annotation by 2 Arabic native speakers and 1 expert linguist	Yes	Newswire	Adj	Yes	No	No	On request
SIFAAT [18]	Yes	3,325	Pos, Neg, Obj	Manual annotation by 2 Arabic native speakers and 1 expert linguist	Yes	PATB [46]	Adj	Yes	No	No	On request
Abtulla et al. [83]	No	3,479	Pos, Neg	Manual translation and expansion with synonyms	Yes	SentiStrength [84]	All	Yes	No	Yes	No
Mourad & Darwish [91]	No	Not reported	Pos, Neg, Obj	Automatic approach using graph reinforcement applied on MT tables	Yes	ArabSenti [77], MT tables from Moses [119]	All	Yes	Yes	No	No
Alhazmi et al. [17]	Yes	10K	Scores for Pos, Neg, Obj	Automatic Mapping of AWN 2.0 to SWN 3.0	No	AWN 2.0	All	Yes	No	No	No
Abtulla et al. [99]	No	16,800	Pos, Neg	Semi-Automatic: manual translation of 300 seed words from SentiStrength and adding synonyms; Machine translating SentiStrength terms and extracting words from annotated Corpus	Yes	SentiStrength, Corpus	All	Yes	Yes	Yes	No
ArSenL [36]	Yes	28,760	3 scores for Pos, Neg, Obj whose sum = 1	Automatic mapping of AWN 2.0 to SWN 3.0 union gloss-synset string matching between SAMA and SWN	Yes	AWN 2.0 [20], SAMA [21]	All	Yes	No	No	Yes
Mahyoub et al. [94]	Yes	7,576	Score for Pos, Neg, Obj	Expansion starting from seed list using AWN 2.0 synset relations	Yes	AWN 2.0	All	Yes	No	No	No
SANA [100]	No	224,564	Pos, Neg, Obj	Manual annotation and automatic expansion through machine translation and a statistical method based on PMI	No	SIFAAT, Yaloo Maktoub, SAMA, Affect Control Theory Lexicon, Tharwa, Twitter, SWN, Youtube General Inquirer	All	Yes	Yes	Yes	No
Al-Ayyoub et al. [105]	No	120,000	Pos, Neg, Obj	Collecting stems and crawling the web for expanding coverage. Translating terms to English and finding corresponding sentiments in English	Yes	Aljazeera.net	All	Yes	No	No	No

Table 2.1: Summary of Arabic Sentiment Lexicons Part 1.

Lexicon	Lemmas	# Entries	Labels	Method	Tested for SA	Source	POS Tags	MSA	Dialect	Emoticons	Available
El Sahar & El-Bethagy [110]	No	6,708	Pos, Neg	Selecting unigrams and bigrams of annotated datasets that correspond to highest and lowest positive and negative coefficients when used in SVM classification.	Yes	Reviews of movies, products hotels and restaurants (Souq, Qaym, TripAdvisor, Elcinemas.com)	All	Yes	Yes	No	Yes
ArSeLex [107, 108]	No	5,244	Pos, Neg, Obj	A list of 400 adjectives is manually annotated and then automatically expanded by checking synonyms/antonyms derived from translations and dictionaries	Yes	Tweets & online dictionaries	Nouns, Adj	Yes	Yes	No	Yes
SLSA [96]	Yes	34,821	3 scores for Pos, Neg, Obj	Automatic gloss-synset matching between AraMorph English gloss terms and SWN synset terms adjusted with heuristics and manual back-offs	Yes	SWN 3.0 [8] and AraMorph [97]	All	Yes	No	No	Yes
NileULex [85]	No	5,953	Pos and Neg	Automatically collected terms and compound phrases from social media postings and manually annotated them; Single word terms were chosen to be as unambiguous as possible and compound words were used to overcome ambiguity	Yes	Social media	All	Yes	Yes	Yes	No
Arabic Senti-Lexicon [111]	Yes	13,760	Manual scoring (-5,+5). Automatic scoring for Pos & Neg (sum = 1)	Translation of MPQA using Google translate followed by manual adjustment; Term expansion using synonyms. Labels/scores added manually by 3 annotators, and automatically using PMI between terms and seed words in a large set of reviews	Yes	MPQA [92]	All	Yes	Yes	No	No
AraSenti-Trans/PMI [112]	No	132K/94K	Pos; Neg; Obj/ Score	Automatic generation using gloss matching against English sentiment lexicons / Computing PMI using automatically annotated Twitter dataset	Yes	Twitter; MADAMIRA [57], MPQA [92], Liu Lexicon [113]	All	Yes	Yes	No	Yes
Assiri et al. [118]	No	14,000	Pos; Neg; Obj	Manual annotation of Saudi Dialect terms in addition to entries from AraSenL obtained after removing diacritics and punctuations from its lemmas	Yes	Saudi dialect and AraSenL [36]	All	Yes	Yes	No	No

Table 2.2: Summary of Arabic Sentiment Lexicons Part 2.

punctuations and diacritics from lemmas. Tables 2.1, 2.2 summarize the important characteristics of some of the sentiment lexicons presented above. In summary, similar to Arabic NLP tools, a good amount of work has been invested to develop sentiment lexicons for MSA but further efforts are needed for DA. Moreover, although some efforts were put to develop large scale Arabic sentiment lexicons, more energy is needed to achieve a similar scale with good accuracy compared to English.

2.2.2 Work on Emotion Lexicons

In this section, we describe the efforts towards creating emotion lexicons. Given the limited number of Arabic emotion lexicons, we cover multiple techniques for creating emotion lexicons for different languages. While sentiment is usually represented by three labels namely positive, negative or neutral, several representation models exist for emotions such as Ekman representation [120] (happiness, sadness, fear, anger, surprise and disgust) or Plutchik model [121] that includes trust and anticipation in addition to Ekman’s six emotions. Despite the efforts for creating large scale emotion lexicons for English, the size of existing emotion lexicons remain much smaller compared to sentiment lexicons.

Strapparava et al. in [122] 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.

[123] 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 [124], WordNet Affect and the General Inquirer [125]. 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 [126]. 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 [127], part of the SenticNet project, includes also around 10K terms extracted from ConceptNet [128] 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 [129] and DepecheMood [130] include words that have emotion scores instead. Affect database extends SentiFul [131] 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 [132].

Bandhakavi et al. worked on constructing emotion lexicons using Tweets annotated with emotion labels [133, 134]. 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 [135] and utilized a feature based supervised approach for classifying emotion.

While the above emotion lexicons were mainly developed for English, [136] 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.

[137] 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.

[138] presented Feel, an emotion and sentiment lexicon for French. Abdaoui et al. utilized NRC emotion lexicon [126] 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.

2.2.3 Work on WordNet Expansion

In this section, we present a brief literature review about existing work related to automatic WordNet expansion for Arabic as well as other languages. Having a large scale hierarchical lexicon where words are semantically linked help greatly in several NLP applications. For example, English WordNet had been used successfully for question answering [139, 140], improving internet searches [141], word sense disambiguation [142], measuring semantic relatedness [143], and last but not least developing large scale sentiment lexicons such as SentiWordNet [8, 9, 144].

Rodríguez et al. in [145, 146] presented two semi-automatic approaches for AWN extension. One approach consists of first automatically deriving translations for corresponding English WordNet synsets, and then, these translations are manually validated by lexicographers. The second approach consists of deriving new derivational Arabic forms from existing synsets in AWN and suggesting potential corresponding English synsets. Again, the suggestions need to be validated manually by lexicographers. Alkhalifa and Rodríguez in [147, 148] presented a semi-automatic approach for expanding specifically named entities coverage in AWN by harvesting existing Arabic information resources about toponyms, countries' name in Arabic and a bilingual Arabic-English named entities lexicon. They also make use of Wikipedia English and Arabic articles and they try to establish links between English and Arabic named entities for automatic expansion. Despite automatically generating named entities suggestions, the approach also involves manual validation. Moreover, the expansion is specific towards named entities. Similarly, Abouenour et al. in 2013 [149] presented an approach to overcome the shortcomings of coverage and usability of AWN. They proposed a semi-automatic approach for expanding named entities in AWN by using YAGO and linking named entities to AWN synsets. They evaluate their expanded AWN in a question answering task. Aminian et al. in [150] proposed an automatic verification and augmentation of multilingual lexicons, specifically THARWA [101] and BabelNet [151, 152]. The authors utilize parallel corpora, monolingual dictionaries and cross-lingual embeddings. Badaro et al. [36] developed the first publicly available large-scale Arabic sentiment lexicon, ArSenL, by combining the result of two sub-approaches. The first sub-approach consists of mapping AWN 2.0 to English SentiWordNet 3.0 (ESWN) [144, 9] and standardizing AWN lemmas to LDC format. The second sub-approach

consists of performing gloss matching between SAMA lemmas' gloss terms and ESWN synset terms, i.e., EWN synset terms. Although ArSenL is a sentiment lexicon, the authors did not only publish the sentiment scores along with the Arabic lemmas, but they also clearly showed the corresponding EWN synset. While not directly targeting WordNet expansion, the works in [153, 154, 155] developed a sense similarity measure by learning low-dimensional latent semantic vectors of concept definitions. While the similarity concept was used for subgroup detection in discussions as discussed in [156, 157], it can also be used for mapping between resources.

In addition to work related to AWN extension, several techniques have been proposed for expanding WordNets in other languages. Bond and Foster in [158] presented a multilingual WordNet where they link all publicly available WordNets together. They also add non Wordnets terms by extracting data from Unicode Common Locale Data Repository (CLDR) and Wiktionary and evaluating similarity of new words with existing senses in WordNet. Similarly, Hanoka and Sagot in [159] make use of Wikipedia and Wiktionary data to align translations between English and French using existing WordNets. They use a back translation algorithm to expand the free French WordNet and they evaluate their approach by computing precision, recall and F1 measure. Sago and Fišer in [160] worked on extending Sloven and French WordNets by recycling existing bilingual dictionaries and encyclopedias and assigning new words to existing synsets based on a classifier trained on several features and similarity measures. Bond and Foster in [158] proposed to extend and link multiple WordNets in different languages including Arabic. For extending WordNets, the authors utilize Wiktionary data in different languages and compute multiple Jaccard similarity scores between Wiktionary data and WordNet synset terms. They manually evaluate their approach by examining 551 alignments. 136 were correct, 48 were considered close enough to produce correct translations and the rest was incorrect. While in the previous discussed work, the goal was mostly to extend an existing WordNet, in [161, 162], the authors proposed to link two existing WordNets without extension. Patel et al. in [162] proposed an approach to link Hindi WordNet to English WordNet using word embeddings. Synsets in each resource are represented by the average of the word vector representations of the corresponding synset terms. Based on existing links between the Hindi WordNet and English WordNet, a translation vector is learned to reduce the distance between the vector representations of a synset in both languages. The translation vector is then used to predict new links. As evaluation metric, the authors utilized accuracy@n measure, where they report the accuracy of predicting a correct link out of top n results returned. They achieve a 0.6 accuracy@10. They claim that their technique can significantly reduce the amount of manual effort required by the lexicographers. Joshi et al. in [161] also worked on automatically linking wordnets between Hindi and English using two sets of heuristics. The first set is based on

bidirectional dictionary from Hindi to English and the second set is based on properties available in WordNets and utilized to compute scores from existing links between English and Hindi WordNets. Using the second set of heuristics proved to be better.

To summarize, most of the proposed approaches for WordNet expansion rely on harvesting existing bilingual data and trying to compute similarity scores with existing WordNet synsets. For Arabic, most of the approaches proposed require manual validation, except for ArSenL, and no benchmark data is presented for automatic evaluation of AWN extension. Thus, we present a benchmark dataset that can be used by researchers for automatic evaluation of AWN extensions and we propose a fully automatic approach for AWN extension as described in chapter 4.

2.3 Arabic Sentiment Corpora

Similar to other machine learning tasks, sentiment analysis requires the existence of annotated data. Annotated sentiment corpora will help in training opinion mining models as well as testing the performance of these models. Abdul-Mageed in [77] annotated 2,855 sentences from PATB part 1 corresponding to the first 400 documents [46]. Two college educated Arabic native speakers annotated the data with four possible labels; objective (OBJ), subjective positive (S-POS), subjective negative (S-NEG) and subjective neutral (S-NEUT). The dataset can be used for subjectivity classification as well as sentiment classification. The distribution of the annotations was 1,281 OBJ, a total of 1574 SUBJ, with 491 were deemed S-POS, 689 S-NEG, and 394 S-NEUT. The dataset is available on request. Abdul-Mageed and Diab in [163] expanded the initial corpus by annotating 5,342 sentences from 30 Wikipedia talk pages and 2,532 sentences from threaded web forums. In total, AWATIF consisted of 10,729 sentences annotated for sentiment. Rushdi-Saleh et al. in [164] presented a corpus consisting of 500 movie reviews in Arabic equally divided between positive reviews and negative reviews. The data was extracted from different web pages and blogs. Aly & Atia in [165, 166] collected a large scale corpus consisting of 63,257 book reviews in Arabic each rated with a scale between 1 and 5. 16,486 users submitted their reviews for 2,131 different books. They utilized the dataset to test different sentiment classification techniques. They also created a sentiment lexicon consisting of single and compound words extracted from the corpus. Saad et al. in [167, 168] proposed an annotation strategy for subjectivity in multilingual setting. Starting with an annotated corpus in English with 5K subjective sentences and 5K objective sentences, the authors built an English sentiment analysis model using trigrams and Naïve Bayes to automatically label English sentences in a parallel English-Arabic corpus. The predicted labels were assigned to the Arabic text as well. Several

parallel corpora were used and a total of 148K sentences were automatically annotated to be either subjective or objective. A sample of 330 sentences was manually annotated to evaluate the automatic annotation and an average F-measure of 68.83% was obtained with an average accuracy of 68.8%. Elnagar and Einea in [169] presented BRAD, a large-scale book reviews dataset in Arabic. The dataset includes 510,598 book reviews provided by 76,530 users on 4,993 books. Each review is annotated with a scale from 1 to 5. A cleaned and balanced version of 156,506 reviews was also released. Several opinion mining models were tested to create baseline results for future research. As an extension to BRAD corpus [169], Elnagar et al. in [170] released a large dataset of hotel reviews written in Arabic and extracted from Booking.com. HARD includes 490,587 hotel reviews collected from the Booking.com website. Each record contains the review text in the Arabic language, the reviewer’s rating on a scale of 1 to 10 stars, and other attributes about the hotel and the reviewer. The dataset includes both MSA and dialects. The authors made publicly available the full unbalanced dataset as well as a balanced subset and published some baseline results on the dataset. Farra et al. in [171] worked on annotating an Arabic corpus for sentiment as well as for the targets of the sentiment in a two-stage process using crowdsourcing via Amazon Mechanical Turk. Task 1 involved identifying the main named entities in a list of comments. Three annotators worked on the task and the intersection of their annotations was selected. The second task was to find the sentiment (positive, negative, neutral) towards the entities derived in Task 1. The dataset was selected from QALB [172, 173] and consisted of 1,177 comments of three different domains: Politics Culture and Sports. They annotated a set of 4,345 Targets: 43% of them had a positive sentiment and 57% had negative opinions about them. The corpus is publicly available.³ HAAD (Human Annotated Arabic Dataset) [174] consists of 1,513 book reviews extracted from LABR and manually annotated for: aspects and their corresponding sentiments with four class labels (positive, negative, conflicting and neutral), aspect categories and the corresponding aspect category sentiment. HAAD included 1,296 distinct aspect terms. Al-Sarhan et al. in [175] worked on developing models for aspect extraction and sentiment classification towards those aspects similar to the subtasks presented at SemEval-2014 Task 4 [176]. For this purpose, they extracted data from Facebook and Twitter related to the 2014 attacks on Gaza and manually annotated them for aspects, aspects’ categories and the corresponding sentiments towards the aspects and towards their categories. In total, they collected a set of 2,265 news posts divided among three sentiment classes: positive, negative and neutral. In total, 9,655 aspects were extracted belonging to four categories: Plans, Results, Peace and Parties. They ran simple sentiment classification techniques on their annotated corpus to set baseline

³www.cs.columbia.edu/noura/Resources.html

results for future research. In [116], the authors collected a set of 8,868 Arabic tweets and manually annotated them for subjectivity and sentiment with five labels: polar, positive, negative, neutral and mixed. They also annotated the corpus automatically with a variety of linguistically motivated features: morphological, syntactic, semantic, stylistic and social signals. ElSahar & El-Beltagy in [110] extracted around 33,116 Arabic reviews about movies, hotels, restaurants and products. They collected the reviews from several websites such as Elcinemas.com, Souq.com and tripadvisor.com. They made the dataset publicly available. While the previous datasets were mainly focused on MSA, Al-Kabi et al. in [177, 178] developed an Arabic sentiment corpus covering MSA as well as several Arabic dialects. The corpus consists of 250 topics and 1,442 reviews extracted from five domains: Economy, Food-Life style, Religion, Sport, and Technology. The corpus was built manually to ensure a high accuracy in terms of annotation. Siddiqui et al. in [179] developed a multifaceted, multilingual corpus for hierarchical sentiment analysis. The different facets included hierarchical nominal sentiment labels, a numerical sentiment score, language used in the review, and the dialect. The annotated corpus consisted of 191K reviews of hotels in Saudi Arabia. The reviews were divided into eleven different sentiment categories ranging from exceptional to very poor. The corpus contains 1.8 million tokens. Reviews were mostly written in Arabic and English but there were instances of other languages as well. A twitter corpus written mostly in Saudi dialect and consisting of 4,700 tweets was also annotated for sentiment by Assiri et al. in [180]. Duwairi et al. in [181] annotated a corpus of 3,206 Arabizi tweets using crowdsourcing. Emerging work handling Arabizi is also presented by Guellil et al. in [182]. Nabil et al. in [115] created ASTD (Arabic Sentiment Tweets Dataset) consisting of 10,006 tweets classified as positive (799 tweets), negative (1,684), mixed (832) and objective (6,691). They investigated different statistics on the dataset and presented a set of benchmark experiments for baseline comparisons. Medhaffar et al. in [183] manually annotated an Arabic corpus written in Tunisian dialect consisting of 17K comments extracted from Facebook. TSAC (Tunisian Sentiment Analysis Corpus) consists of almost equal number of positive and negative comments. The authors tested several sentiment analysis models on the developed corpus. Recently, AraSenti-Tweet [114] was developed and made publicly available at the end of 2017. AraSenti-Tweet includes 17,573 tweets mainly covering Saudi dialect and annotated with four sentiment labels: positive, negative, neutral and mixed. Benchmark and baseline results are also provided for easier research use and comparison. MIKA [184] is a multi-genre tagged corpus for MSA and DA. MIKA was manually collected and annotated at sentence level with semantic orientation (positive or negative or neutral). A number of rich set of linguistically motivated features such as contextual intensifiers, contextual shifters, negation handling and syntactic features for conflicting phrases were used for the annotation process. The data consists of MSA and Egyptian DA

from different sources such as Twitter, hotel reviews, product reviews and TV programs reviews. MIKA includes 4,000 reviews annotated for sentiment. ArSAS [185] is an Arabic corpus of tweets annotated for the tasks of speech-act recognition and sentiment analysis. A 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: positive, negative, mixed and neutral. ArSentD [186] is 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. The authors claimed that the results confirm the importance of these annotations at improving the performance of a baseline sentiment classifier. They also claimed that ArSentD helps in closing the gap of training in a certain domain and testing in another domain. Hamdi et al. in [187] annotated a set of 15,274 reviews extracted from online reviews, Facebook comments and Twitter posts related to governmental services. The annotation was multifaceted and was not only limited towards identifying the sentiment polarity but also domains, dialects and linguistic issues. Annotators were from different Arab countries and they were trained for the annotation task. In [188, 189], Itani et al. extracted a corpus from Facebook comments consisting of DA. The corpus was manually annotated into five sentiment labels: positive, negative, neutral, dual and spam. The corpus included 2K comments in total distributed among the 5 classes. In [190], Al Mukhaiti et al. extracted a dataset from multiple social network websites namely Youtube, Twitter, Facebook, Instagram and Keek. After filtering the data, a total of 2,009 reviews were annotated to be either positive (1,004) or negative (1,005). While previous efforts were mainly about annotating Arabic texts, and similar to [167, 168], Elnagar et al. in [191] studied the impact of translating resources from English to Arabic to perform sentiment analysis on Arabic text. They claimed that although the translation might not be accurate grammatically, sentiment was preserved in most of the cases due to main keywords. In 2018, Ahmed [192] developed a large-scale Arabic sentiment corpus (GLASC) consisting of online news articles and metadata. GLASC was annotated for both sentiment labels (positive, negative, neutral) and sentiment scores between -1 and +1. GLASC consists of around 620K news articles. Ahmed also worked on translating the English SenticNet [193] to Arabic and creating ArSenticNet which includes 48K Arabic concepts. The author experimentally tested multiple sentiment classification approaches at document level and sentence level with the best models using fusion between SVM and HMM for documents and SVM and Linear regression for sentences.

Development of Arabic sentiment lexicons and corpora has gained interest among researchers. In fact, Arabic sentiment corpora are currently of comparable sizes compared to English such as for example the balanced version of BRAD [169]

with more than 156K reviews. Most of the datasets created were developed in-house and several were not made publicly available. It is crucial to share these resources publicly to help accelerate research work in the domain. It would also be beneficial to have a unified platform where these resources can be found to increase visibility to researchers interested in Arabic opinion mining.

After discussing different aspects of Arabic resources covering tools, lexicons and annotated sentiment corpora, we describe in the next section state-of-the-art opinion mining approaches that utilize extensively these resources.

2.4 Opinion Mining Approaches

Over the last two decades, several approaches have been adopted to perform sentiment analysis (SA) in different genres of English text (reviews, messages and micro-blogs), and have reached satisfactory performances [194, 195, 196]. Sentiment models have benefited from available NLP resources in English such as POS taggers and grammatical parsers, as well as lexical resources including sentiment lexica such as SentiWordNet [8] and MPQA [92] and sentiment-labeled corpora such as the Stanford Sentiment Treebank [10].

These resources are often scarce or nonexistent in other low-resource languages, which poses serious obstacles to performing accurate sentiment analysis in these languages. Many of these resources have only recently been developed for Arabic, as discussed in Sections 2.1 and 2.2, which opened doors for extensive efforts to explore the application of sentiment analysis in Arabic text. Early efforts included manual attempts by the use of conceptual model to represent opinion as described in [197]. This section discusses approaches and highlights the progress made at developing sentiment analysis solutions in Arabic.

Sentiment analysis approaches can be categorized, at the macro-level, into *lexicon-based* and *supervised learning* approaches.

Lexicon-based approaches mainly depend on algorithms that use sentiment lexica to predict the sentiment, whereas supervised learning approaches are modeling algorithms trained on labeled examples to learn complex relations between features extracted from the texts and the associated sentiment labels. Supervised models can be further categorized into models based on *feature engineering* and others based on *deep learning*. Feature engineering-based approaches predict sentiment by learning from a variety of features that are selected to capture different aspects of the text. On the other hand, deep learning is considered the state-of-the-art of machine learning, and has achieved significant success in many domains, mainly computer vision and NLP [198]. It generally uses embedded representations of text units (characters or words) as input features to train different neural networks architectures.

Recently, combination of multiple approaches have been used for Arabic

sentiment classification for Twitter data [199, 200, 201, 41] and online reviews [202, 203]. For instance, Refaee and Rieser in [204] participated in SemEval 2016 Task 7 [88] detecting sentiment intensity of English and Arabic phrases. The authors proposed a hybrid approach that consists of a supervised model that uses ensemble of trained linear regression models followed by a rule-based approach with sentiment lexica to adjust the predicted intensities. Refaee and Rieser were able to achieve the first rank in this task with a 0.536 Kendall score. Such models can also be observed across the systems participating in SemEval 2017 Task 4: Sentiment Analysis for Twitter [205]. Five subtasks were dedicated for Arabic Sentiment Analysis in Twitter data. The tasks were as follows: subtask A: given a tweet, classify it as either positive, negative or neutral; subtask B: given a tweet and a topic, decide whether the tweet is positive or negative; subtask C is similar to subtask B but there are five sentiment classes: strongly and weakly positive/negative and neutral; subtask D: given a tweet and a topic, estimate the distribution of tweets across the positive and negative labels, and last but not least, subtask E is similar to subtask D but with five different sentiment labels as in subtask C. Below, we provide for each category a detailed description of the main sentiment approaches that were developed and were applied to Arabic.

2.4.1 Lexicon-based Approaches

Sentiment lexicon-based approaches are to some extent simple and usually unsupervised. They do not need large-scale and expensive sentiment-labeled datasets that are used to train supervised machine learning models. However, these approaches require external resources, mainly sentiment lexica, to predict sentiment. One of the most widely used lexicon-based unsupervised algorithms predicts the sentiment of a text by accumulating the scores of its words, based on some lexicon, and then checking the sign of the resulting score. Usually, scores of +1, -1 and 0 were assigned to positive, negative and neutral words, respectively. Alternatively, numerical scores indicating sentiment intensity were also used when available.

Awwad and Alpkocak in [206] evaluated the performance of four different sentiment lexicons derived from English sentiment lexicons Harvard IV-4 Dictionary and MPQA. Awwad and Alpkocak obtained similar performance on different datasets and concluded that lexical based approaches for sentiment analysis have similar performances. Hamdi et al. [187] proposed CLASENTI a class-specific sentiment analysis framework. CLASENTI includes a lexicon-based model to calculate polarity strengths. The annotated lexicon is filtered based on the specific classes of the domain and dialect. Ahmad et al. in [207] developed a general approach for sentiment analysis from Financial data streams by applying their English approach to Chinese and Arabic financial texts. By identifying keywords specific to financial news, the authors

looked at the neighboring terms of those keywords to identify the polarity by using a sentiment lexical resource. Mohammad et al. in [208] experimented with different available sentiment lexicons to perform sentiment classification on Arabic text extracted from social media. They used existing Arabic sentiment lexicons and translated ones from English to Arabic. A combination of an Arabic Dialectal hashtag lexicon [209] and a translation of NRC Emotion lexicon achieved best results. Elhawary and Elfeky in [210] utilized an unsupervised approach to perform sentiment classification on Arabic Business reviews. Using a sentiment lexicon and a set of rules that detect intensification and negations, each sentence was labeled with a total score that defined as a result the polarity of the sentence. In [211, 212], Al-Subaihin et al. proposed an unsupervised technique to perform sentiment analysis on Arabic dialectal text at the sentence level. They utilized an unsupervised lexicon-based approach to try to overcome the challenge of limited resource availability for dialects. They used human computing which is a technique that integrates human effort to solve certain steps in a system. To keep the sentiment lexicon enriched, the authors built an online game where players help in annotation of the lexicon. The lexicon was used to assess the polarity of a given sentence. Siddiqui et al. in [213, 214] presented a sentiment analysis system that is based on a set of handcrafted rules extracted with the help of sentiment lexicons such as text begins with, ends with, includes or is equal to. They tested their technique on two different corpora achieving accuracies of 93.9% and 85.6%.

In [99], the lexicon-based algorithm was applied to MSA comments using a sentiment lexicon of 16,800 words that was constructed automatically and manually. In [215], the same approach was applied to tweets and comments, written in both MSA and DA, using a sentiment lexicon of 4,800 words that was created by expanding a seed of 300 words using synonym and antonym relations. Experimental results indicate that light-stemming degraded the performance. Furthermore, in [105], a lexicon-based algorithm was applied to tweets, written in both MSA and DA, using a sentiment lexicon of 120,000 words that was constructed by translating existing lexica in English. Other works that adopted this algorithm include [83] and [216]. Improvements to this algorithm were explored by [217] to account for negations by adding hard-coded rules derived after extensive analysis of negation forms in Arabic. The algorithm was used along with ArSenL lexicon [36]. Also, the work in [218] proposed a rule-based parser for document segmentation, and then used an Arabic translated version of the MPQA lexicon [92] to aggregate the sentiment scores over the document taking into consideration negations, intensifiers and conjunctions. Similarly, Bayoudhi et al. in [219] proposed to develop models for Arabic sentiment classification at the sub-sentential level which could provide very useful trends for information retrieval and extraction applications, Question Answering systems and summarization tasks. They started by (1) building a high coverage sentiment lexicon with a semi-automatic approach; (2)

creating a large multi-domain annotated sentiment corpus segmented into discourse segments in order to evaluate the sentiment approach; and (3) applying a lexicon-based approach with an aggregation model taking into account challenging linguistic phenomena such as negation and intensification. Obaidata et al. in [220, 221] proposed a lexicon-based approach for aspect extraction and classification of sentiments towards those aspects. They experimented on a dataset HAAD [174]. The authors utilized HAAD to automatically generate an Arabic sentiment lexicon using the frequency of occurrence of the term with a specific polarity. When using the lexicon in a sentiment analysis task, words that were not in the lexicon were translated to English and sentiment scores were obtained from SentiWordNet. Matoui et al. proposed in [222] a lexicon-based approach for sentiment analysis from dialectal Arabic text, mainly the Algerian dialect. The Algerian dialect includes a lot of code switching specifically switching between Arabic and French. They showcase the challenges present in the Algerian dialect and develop techniques to overcome them. They also develop three sentiment lexicons manually. The first one is based on existing sentiment lexicon for Egyptian dialect where the authors only kept terms that are common with Algerian dialect. The second lexicon is a list of negation words that are frequently used in Algerian dialect and third one is a list of intensifiers also frequently used in Algerian dialect. They tested different configurations for their model. The first one works at the phrase level and computes similarity of a given comment with existing labeled phrases. The second configuration involves going to the word level after the application of their own developed parser for Algerian dialect to perform tokenization, stopwords removal and normalization. The tokens are then processed by a language detection and stemming module that identifies the language of the tokens. For Arabic tokens, stemming was applied while tokens in other languages were first translated to Arabic and then stemmed. Stems were matched against the developed sentiment lexicons in order to compute a text semantic orientation score. They collect and manually annotate for polarity a set of 7,698 Facebook comments that cover multiple topics and include MSA and Algerian dialect. They achieved best accuracy of 79.13% when combining the two system’s configurations. Recently, in SemEval 2017 Task 4 [205], Mulki et al. in [223] applied an unsupervised lexicon based approach for Arabic message polarity classification in subtask A. As sentiment lexicons, they have utilized NileULex [85], Arabic Emotion lexicon for Emojis and Arabic Hashtag Lexicon for MSA and DA [224, 225]. They have also built two manually annotated sentiment lexicons to cover Levant and Gulf dialects. Specific efforts towards Emirati dialect were also seen in [226]. In [227], Htait et al. also implemented an unsupervised sentiment classification model using word embeddings and sentiment lexicon extracted from annotated Arabic Tweet corpora [115]. The sum of cosine similarity measures between the vector representation of the tweet and the words in the sentiment lexicon is used for

message polarity classification. This technique placed Htait et al. team, LSIS, in 5th rank among 8 participants in subtask A.

2.4.2 Feature Engineering “Supervised” Approaches

With the availability of NLP tools and several lexical resources, a wide range of features were made available through feature engineering to train supervised machine learning models for opinion mining, achieving better performances compared to the lexicon-based models. The aim of extracting these features is to reflect the different aspects of the given text. Thus, these features vary in complexity depending on the information they convey. According to [228], optimal set of feature for opinion mining is dataset dependent. They typically range from shallow features that capture the surface form of the text, to syntactic features that capture the grammatical rules that govern the language construction, to deeper semantic features that capture the underlying meanings of the different components of the text.

Surface features Mainly include word n -grams; sequences of n consecutive words. These features were extensively evaluated under different settings including different context lengths n and different feature representations: binary presence, term frequency (TF) and term frequency inverse document frequency (TFIDF) [229]. These features were used to train several machine learning algorithms for classification, mainly SVM, Multinomial Naïve Bayes (MNB), Conditional Random Fields (CRF), Decision Trees and k -Nearest Neighbors (k -NN). Overall, in some cases, SVM achieved better results [164, 165, 230, 231, 232, 233, 234, 235, 217, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251], and in other cases, NB performed better [252, 253, 254, 255, 256, 257] especially in the case of unbalanced datasets such as in [258, 259, 260]. Mostafa in [261] claimed that the best classifier is dataset dependent. Ensemble classifiers achieved further improvements [262, 263, 264, 265, 266]. In order to handle the impact of out-of-vocabulary (OOV) words due to dialectal variations, a lexicon was manually created in [267] to map dialectal words into their corresponding MSA forms. The use of this lexicon introduced a slight improvement. Azmi and Alzanin presented Aara' [254], a system for mining opinion polarity through the pool of comments that readers write anonymously at the Online edition of Saudi newspapers. N-grams were used as features with a Naïve Bayes classifier and the output consisted of four class labels: positive, negative, neutral and mixed. For training, they manually marked the comments as belonging to one of the four categories. All the words in the documents of the training set were removed except those with explicit sentiment connotations. The system carried out polarity classification over informal colloquial Arabic that is unstructured and with a reasonable proportion of spelling errors. The result of testing the

system showed a macro-averaged precision of 86.5%, while the macro-averaged F-score was 84.5%. The accuracy of the system was 82%. Al-Obaidi and Samawi in [268] worked on developing sentiment classification model for DA specifically Jordanian and Saudi dialects. They contributed to the preprocessing part of the system by developing a tailored stop words list for the two dialects and a light stemmer specific for the dialects. They experimented with different classification techniques along with bag of words and n -gram features. Maximum Entropy performed best with trigrams. Results in these papers indicate that there is no consensus regarding the best experimental setup (length of n -gram, or representation), and that results are corpus-dependent. This is expected given the simplicity of these features, which does not match the complexity of the task. According to Halees [269], word embeddings representation performed better in terms of accuracy on four different datasets compared to when using n -grams with the same classification model. Badaro et al. in [44] also found word embeddings extracted from AraVec [76] to perform better than other features with ensemble classification. The same conclusions were drawn in [270, 271]. Similarly, Al-Azani et al. in [272] observed that simpler models consisting of word embeddings and emojis performed better than when using typical n -grams. El Razzaz et al. in [273] evaluated the use of Arabic word embeddings. Additional surface features, also referred to as stylistic features, were extracted to indicate lexical and structural style markers. Character n -grams improved the classification performance significantly when combined with word n -grams to classify political articles using SVM and k -NN [274] and to classify newspaper articles [275]. Other stylistic features including digit n -grams, word length distributions, vocabulary richness measure, function words and punctuation were also combined with word n -grams to train an SVM classifier, and performed well after applying the Entropy-Weighted Genetic Algorithm (EWGA) selection method [276]. Recently, TwStAR [223] tried using unigrams, bigrams and trigrams with SVM to perform sentiment classification on Arabic Twitter data in SemEval 2017 Task 4. They achieved rank 7 among 8 participating teams in subtask A.

Syntactic features Used to reflect the structure of the text and understand how the words function and combine together to express meaning. Knowing that Arabic is a morphologically rich and complex language, it is of high importance to incorporate syntactic and morphological information of the language into the sentiment models. One of the earliest grammatical approaches proposed to generalize verbal and nominal phrases into one form based on ‘actors’ and ‘actions’, and then to train SVM using the following features: actors, actions, adjectives, nouns, syntactic type of sentence, conjunction with previous sentence, and word sentiment polarity [277]. Most of this information was manually labeled due to the lack of Arabic NLP tools at

that time. Results are considered a ‘proof of concept’ of the importance of incorporating syntactic information into the model. The recent establishment of advanced Arabic NLP tools and resources allowed the automatic extraction of syntactic and morphological features, which are used to mitigate the impact of complex morphology on sentiment. Examples of such resources include the Arabic Treebank (ATB) [46], the Standard Arabic Morphological Analyzer (SAMA) [21], MADAMIRA [57] and many other tools mentioned in section 2.1. For instance, adding word-level inflectional morphological features (gender, number, voice, ...) to basic surface features improved the performance of sentiment classification in MSA data [77], whereas they resulted in performance degradation when applied to Twitter data [278]. This is mainly because most Arabic NLP tools are trained on MSA data, whereas tweets generally contain significant amounts of dialects and misspellings, thus resulting in error-prone features. One way to alleviate the complexity of concatenative morphology in Arabic words was to represent words by stems or lemmas. Stemming was widely used as a preprocessing step, where all clitics and suffixes were chopped off from the base words, whereas lemmatization refers to selecting one word form to represent a set of words that are related by inflectional morphology [13]. It has become conventional to extract stem, root or lemma n -grams, instead of raw word n -grams, and use them to train sentiment classifiers [77, 83, 236, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289]. Both forms provided the capability of generalizing to new unseen words that are morphologically related to words in the training data. However, it was not clear which form performed better. Furthermore, the impact of augmenting lemma n -grams with part-of-speech (POS) tags was measured for both tasks of subjectivity analysis and sentiment analysis. It was found that POS tags do not provide further improvements for sentiment analysis, whereas they do slightly improve subjectivity analysis [280]. Abd-Elhamid et al. in [290] used rules and syntactic features for sentiment classification on Arabic text using decision trees. Al-Smadi et al. in [291, 292] presented an aspect-based sentiment analysis approach using a feature engineering approach. As features, they considered POS tags, named entity recognition and n -grams with n varying between 1 and 3. They experimented with different classifiers: CRF, k -NN, Decision Trees and Naïve Bayes. They extracted 2,265 news posts from Al Arabiya and Al Jazeera. The dataset was annotated by three Arabic native speakers. The annotators had to extract aspect terms and had to assess their polarity: positive, negative or neutral. Preprocessing included tokenization, segmentation, stemming, POS tagging, punctuation and stop words removal and normalization. For the task of extracting aspects, decision trees performed best with F1-measure of 81.7% while for the task of sentiment classification CRF performed best with an accuracy of 86.5%. CRF were also utilized successfully in [293, 294]. k -NN performed best when used to perform multi-way classification in a set of cascaded classifiers [295, 296]. In [297], the authors presented an approach to

extract and classify opinion in micro-blogs. Their approach is based mainly on linguistic features extracted from Kuwaiti dialect and employed with a SVM classifier. They tested their approach on a corpus of 340,000 tweets about “interrogation of ministers by the National Assembly of Kuwait” during the last two years. Tweets were collected automatically by a module developed in java. The corpus had been manually annotated by three Kuwaiti dialect native speakers. An average value of 76% and 61% were obtained for precision and recall respectively in terms of sentiment classification.

Semantic features Many lexical resources were developed to reflect the semantic aspects of words and phrases in Arabic. Examples include the Arabic WordNet [20], an equivalent of English WordNet [7], which groups words into sets of synonyms (synsets) that are also enriched with semantic relations such as hyponymy, synonymy and antonymy. Additionally, several Arabic sentiment lexica have been developed using automatic, semi-automatic and manual approaches, as described in section 2.2.1. The development of these resources enabled the extraction of more complex sets of engineered features that reflect surface, syntactic and semantic aspects of the texts. Scores from sentiment lexica were frequently used to represent n -gram features, replacing presence and TFIDF scores [35, 37, 298, 299, 300, 42]. Sentence-level sentiment features were also derived by aggregating word scores through averaging or summation, or by generating binary features indicating presence of positive and negative terms in the text. These features were combined with surface and syntactic features to train sentiment classifiers that performed well [77, 91, 36, 163, 280, 301, 302, 303, 304]. In [305, 306], ontology was used to derive features for sentiment analysis from Arabic text reviews. To extract the ontologies present in a review, the author utilized ConceptNet [307] and WordNet [7]. The ontology allows the system to detect the sentiment for each element in the ontology. The technique was applied on two different datasets one for hotels and one for books each consisting of 2,000 reviews equally split between positive and negative. An average accuracy of 79% was claimed to be achieved. Ontologies were also used in [308, 309, 310, 311, 312]. Entries of the ArSeLEX lexicon were used along with a set of linguistically and syntactically motivated features including contextual intensifiers, contextual shifters and negation particles to train a SVM sentiment classifier, and achieved high performances on small Twitter, comments and reviews datasets, written in both MSA and Egyptian dialect [107]. Refaee and Rieser in [313, 314] tested the efficiency of using emoticons for automatically classifying tweets into subjective or objective and also detecting whether the tweet is positive or negative. They found that emoticons perform well for detecting subjectivity however they perform poorly in distinguishing polarity. Same analysis about emoticons was suggested in [315]. Rizkallah et al. in [316] addressed the problem of sentiment

analysis on Twitter data by comparing two different approaches: the first one consisted of applying feature extraction directly on Twitter data in its dialectal form and the second one consisted of first translating the dialectal Arabic to MSA and then applying sentiment analysis model. The authors annotated manually a set of 2,010 tweets written in Saudi dialects into four labels: positive, negative, neutral and mixed. Based on their results, translating to MSA seemed to improve the overall performance of sentiment classification with an accuracy of 76.2% when using logistic regression. In [317], Al-Harbi et al. also worked on developing sentiment analysis model for Saudi dialect. They studied the effect of different preprocessing techniques on the performance of sentiment classification. They used a dataset of 5,484 Saudi tweets annotated for sentiment. The preprocessing included no stemming, stemming, light stemming or replacing Saudi dialect terms with their corresponding MSA term. Using the latter showed to perform best along with k -NN as classification technique. Mustafa et al. in [318] proposed an improvement over the typical Bag of Words (BoW) model for sentiment analysis by incorporating different feature sets and performing cascaded analysis that contains lexical analysis, morphological analysis, and semantic analysis. AWN was used to extract semantic relations between terms in the dataset. Moreover, specific feature extraction components were integrated to account for the linguistic characteristics of Arabic. Emoticons and smileys were as well extracted to reflect the nature of the social media content. An average F-measure of 89% was achieved with a claimed significant improvement compared to BoW. El-Naggar et al. in [319] presented a hybrid approach for sentiment analysis of MSA and Egyptian DA using verbal and non-verbal cues in the form of text and emojis. Using Plutchik’s Wheel of Emotions, an emoji lexicon was created as a resource for non-verbal emotion classifications. Their feature set consisted of unigrams and bigrams with a minimum frequency of 5, sentiment score derived based on different syntactic rules, emotion labels, number of sentiment and emotion tokens, presence of negations and count of total number of tokens in the tweet. SVM performed best when used with bagging. The authors achieved an accuracy of 90%. CLASENTI [187] includes a two-stage machine learning sentiment analysis. First, full-corpus (i.e., trained on all the annotated dataset) models classify the incoming text polarity, domains, dialects, and linguistic issues. Second, class-specific models are trained on filtered subsets of the corpus according to the performances of the full-corpus models. Moreover, a set of hand-crafted features, that proved successful in English [320], was adopted in Arabic [39]. This feature set consisted of character and stem/lemma n -grams, counts of punctuation marks, elongated words, negated contexts, positive and negative emoticons, POS tags, and of positive and negative words found in MSA and dialectal sentiment lexica. These features were used to train the model submitted by the OMAM team [38] to SemEval-2017 Task 4 on sentiment analysis on Arabic Twitter dataset. OMAM ranked 1st subtasks C

(*3-way topic-based sentiment classification*) and E (*3-way tweet quantification*). The NileTMRG team [321] trained a Naïve Bayes classifier using the different features described above with additional Twitter specific features such as whether the tweet starts with a hashtag [321]. This system ranked 1st in subtask A with an average recall of 58.3%. The same team implemented an ensemble system for subtasks B and D, consisting of a convolutional neural network (CNN), a multi-layer perceptron (MLP) and a logistic regression (LR) predictor. The input to the CNN classifier was an aggregation of the embeddings of each tweet’s words, where these embeddings are obtained by training Word2Vec [322] using a 4-million Arabic Twitter dataset. The other classifiers were trained using a set of hand-crafted features including: a bag-of-words (BoW) representation of the tweet, the sentiment of the tweet as predicted by the team’s system for subtask A, the number of positive/negative terms, the position of the target topic in the tweet, the presence of positive/negative emoticons, the presence of positive/negative terms around the target topic, the number of positive/negative terms in the first and in the second half of the tweet. This system also ranked 1st for both subtasks. Other participants tried combining features from different approaches to improve the accuracy of their systems. Jabreel and Moreno in [323] trained SVM with a rich set of surface, syntactic and semantic features, in addition to tweets embeddings generated by summing the word embeddings of their constituent words. SiTAKA was ranked 2nd in subtask A (*sentiment classification*). The INGEOTEC team used an ensemble classification system and ranked 4th in subtask A [324]. In this system, the output of a generic sentiment classification (B4MSA) system [325] was combined using the EvoDAG Genetic programming system [326, 327].

Aspect-Based Sentiment Analysis Ismail et al. in [328] proposed an approach for performing aspect-based sentiment analysis on 500 Arabic movie reviews, 1,000 restaurant reviews and 500 reviews of mixed domains. They started by manually annotating a set of Arabic syntactic patterns and roots with sentiment binary scores. They used the manually annotated lexicon to tag the words in the corpus. They also identified intensifiers and negations. They formulated a set of patterns in order to detect automatically aspects. They assigned for each aspect the sum of sentiment scores of the words that describe it. They claimed to achieve an average accuracy of 85.9% in terms of detecting the correct orientation of a given aspect. They also achieved an average F1 measure of 79.1% in terms of accurately extracting aspects. To enable aspect-based sentiment analysis from Arabic texts, Mataoui et al. in [329] proposed a syntax based approach for aspect detection from Arabic reviews extracted from Trip Advisor and Souq.com. Ibrahim and Salim in [330] developed aspect-based sentiment analysis system that takes as inputs an

Arabic tweet and generates as outputs the sentiment of the tweet and the different aspects or entities described in the tweet. Their work targeted DA. Extraction of entities was based on a frequent pattern mining approach. For predicting sentiment, a set of features were generated including POS tags, N-grams and polarity scores from a sentiment lexicon. In [331, 332], Al-Smadi et al. compared the performance of RNN to SVM when using a set of surface, syntactic, and semantic features. The authors claimed that SVM performed better in terms of accuracy but RNN had a faster training and testing time in both prediction of aspects and classification of sentiment towards those aspects. Farra and Mckeown in [333] presented a system that is applied to complex posts written in response to Arabic newspaper articles. Their goal was to identify important entity targets within the post along with the polarity expressed about each target. They claimed to achieve significant improvements over multiple baselines, demonstrating that the use of specific morphological representations improves the performance of identifying both important targets and their sentiment, and that the use of distributional semantic clusters further boosts performances for these representations, especially when richer linguistic resources were not available. Zarra et al. in [334] focused on Maghrebi DA. On a corpus extracted from different Facebook pages, the authors implemented a supervised approach to extract the sentiments, and an unsupervised approach to extract topic. Then, they proposed a semi-supervised approach that combines the topic and the sentiment in a single model, in order to assign each topic to a specific sentiment.

2.4.3 Deep Supervised Approaches

Currently, state-of-the-art artificial intelligence (AI) systems rely on deep learning techniques, and has achieved immense successes in many domains, especially computer vision and NLP. In NLP applications, deep learning benefited from the invent of word embeddings; distributional vector representations that encode syntactic and semantic properties of the words into low-dimensional and dense vectors. Examples of word embedding models include CBOW and Skip-gram models in word2vec [322] and GloVe [335]. Word embeddings proved successful in several NLP applications such as POS tagging, chunking and parsing, word-sense disambiguation, named entity extraction and sentiment target entities [336, 337, 338, 339]. One of the main advantages of deep learning models lies in their ability to perform semantic composition: generating a vector representation for text units by combining their finer-grained constituents or entities efficiently and in a low-dimensional space. Consequently, advanced deep neural network architectures have been successfully applied for English sentiment analysis, such as recursive neural networks [10], deep convolutional networks [340], Gated Recurrent Neural Networks (GRNN) [341], dynamic memory networks [194] and the human

reading for sentiment (HRS) framework [342].

Following their success in English, deep learning models were first used for sentiment analysis in Arabic in [343], where several deep learning models including Deep Belief Networks (DBN), Deep Neural Networks (DNN) and Deep Auto Encoders (DAE) were trained using word n -grams. Although these models proved better than several SVM sentiment classifiers such as in [344], they still suffer from the sparse input features, indicating the importance of word embeddings. Al-Sallab et al., [31] analyzed and identified several limitations that might prevent deep learning models to achieve high performances in Arabic similar to what they did in English. These limitations are mainly related to lexical sparsity, which is due to Arabic rich morphology and complex concatenative system, as well as the usage of non-standardized dialects, leading to large numbers of out-of-vocabulary (OOV) tokens. Also, using traditional word embeddings is sub-optimal for sentiment analysis, since these vectors do not capture sentiment properties of the words. These limitations were addressed in the AROMA model [31], where the performance of the Recursive Auto Encoder (RAE), as used in English [11], was significantly improved by using newly-proposed word sentiment embeddings as input features, and by applying the recursion to morphologically tokenized text (separating clitics from words according to [345]) following the path of phrase structure parse trees. The first Arabic Sentiment Treebank (ArSenTB) was developed in [346] and consists of parse trees, where each node of the tree (corresponding to a word, phrase or full text) is associated with a sentiment label. ArSenTB allowed to train Recursive Neural Tensor Networks (RNTN) that can predict the overall sentiment by using the intermediate sentiment labels at the internal nodes, following the idea proposed in English [10]. Furthermore, this model was enriched with morphological features including stems, lemmas and roots in order to overcome the lexical sparsity and ambiguity issue. These features, especially stems, achieved significant performance improvements on a data containing a mixture of MSA and DA. Convolutional Neural Networks (CNNs) have also been utilized for Arabic sentiment classification using word embeddings [347, 348, 349]. Alayba et al. in [350, 351] applied sentiment analysis on a health dataset using deep learning models specifically deep neural networks, CNNs and Long Short-Term Memory (LSTM) networks. The features they used were: characters, character n -gram and words. Each feature was represented as an embedding vector based on different sentiment analysis levels. Each feature was tested on its own. The input data was fed to CNN followed by a Max pooling layer. The output vectors of the max pooling layer were then used as input to LSTM networks to measure the long-term dependencies of feature sequences. The output vectors of the LSTMs were concatenated, and an activation function was applied to generate the final output: either positive or negative. The deep neural network accuracy reached 85%, while the CNN accuracy was better reaching 90%.

Ruder et al. in [352] presented a hierarchical bidirectional LSTM network for performing aspect-based sentiment analysis on a multilingual dataset. They utilized word embeddings to represent terms and each aspect was represented by an entity and an attribute. Word embeddings were fed into a sentence-level bidirectional LSTM. Final states of forward and backward LSTM were concatenated together with the aspect embedding and fed into a bidirectional review-level LSTM. At every time step, the output of the forward and backward LSTM were concatenated and fed into a final layer, which outputs a probability distribution over sentiments. They evaluated their model on SemEval 2016 dataset [353] that consists of 11 domain-language datasets containing 300-400 reviews with 1,250 to 6,000 sentences. The LSTMs had one layer and an output size of 200 dimensions. The authors used 300-dimensional word embeddings. For Arabic, they learned embeddings using Leipzig Corpora Collection.⁴ They were able to achieve an accuracy of 82.9% for Arabic. Ruder et al. in [354] applied a similar approach but with CNNs for both aspect extraction and aspect-based sentiment analysis on SemEval 2016 dataset and achieved top ranks in multiple languages. In [355], Yu and Al Baadani developed a multilayer perceptron (MLP) sentiment classification model for DA. The data preprocessing consisted of two phases: (1) segmentation phase where a separation between MSA, DA and non-Arabic was performed; (2) refinement phase which included removing elongation effects, normalization and removing diacritics. Features included negation and intensification detection, polarity score assignment for terms and polarity tag assignment for emoticons based on their developed lexicons. As a dataset, the authors collected comments from several social media platforms such as Google plus, AreebaAreeba, Facebook, Youtube, Twitter, Yahoo News and WeChat Moments. They annotated the comments into five sentiment labels varying from highly positive to highly negative. The annotated dataset consists of 14,000 comments. They used an ensemble of 3-layer MLP networks consisting of one hidden layer in addition to input and output layers. The input for each MLP is a random subspace of their proposed features. Using 10-fold cross validation, they were able to achieve an average accuracy of 89.75%. Al-Azani and El-Alfy in [356] tested LSTM models unidirectional and bidirectional, and GRNN. Bidirectional LSTM outperformed typical classification techniques when emojis were used as input to the network. In [357], Al-Azani and El-Alfy evaluated different deep neural network architectures including LSTM, GRNN and CNN for sentiment classification using as input word embeddings. LSTM combined with CNN performed best on two different datasets. LSTM was also combined with CRF successfully in [358] for aspect-based sentiment classification. In [359], Barhoumi et al. utilized Doc2Vec to generate paragraph embeddings for Arabic texts such that the vector representation would be used as input feature for a classifier. In

⁴<http://corpora2.informatik.uni-leipzig.de/download.html>

order to generate the embeddings, they utilized the sentiment annotated corpus LABR [165]. Multilayer perceptron (MLP) and logistic regression were used as classification techniques. They claimed that using light stemming as preprocessing before computing the vector representations improved the performance with MLP. Doc2vec was also employed by Abdullallah and Shaikh in [360]. The authors participated in SemEval 2018 Task 1 Affect in Tweets [361]. They applied the same model across all subtasks for both English and Arabic datasets. They used doc2vec and word2vec to generate word embeddings which were appended to a set of psycholinguistic features. They used a fully connected neural network architecture to train their model. They ranked 4th in both sentiment related tasks.

We can notice that the popularity of deep learning models for Arabic Sentiment Analysis is increasing, where several participants at SemEval 2017 Task 4 [205] employed deep learning models for sentiment classification. For instance, the ELiRF-UPV team [362] proposed a neural network architecture that consists of CNN, MLP and bidirectional long short-term memory (BLSTM) recurrent networks. The first layer of the CNN consists of a unidimensional convolutional layer that allows extracting spatial relations among the words in a tweet. In some subtasks and after the convolutional layer, a down-sampling process is applied using a max pooling layer. The output of the convolutional layer (32–256 neurons) is fed to a BLSTM, which performs semantic composition and generates an output representation that is fed to a fully connected MLP consisting of 1 up to 3 hidden layers (depending on the subtask). A softmax function is then used to estimate the probability of each class. The input to this system was an aggregation of: out-domain word embeddings learned from Wikipedia Arabic articles; in-domain word embeddings learned from the corpus provided by SemEval organizers; and a one-hot vector representing word sentiment polarity derived from NRC lexicon [126, 363]. This system ranked 3rd in subtasks A and D, and 2nd in subtasks B, C and E. One possible interpretation for NileTMRG achieving better results than EliRF-UPV is that the former trained word embeddings on a much larger corpus retrieved from Twitter, which will more likely resemble SemEval dataset, while EliRF-UPV trained word embeddings on Wikipedia, which is somehow different than Twitter data. Also, the results achieved by NileTMRG indicates that while deep learning represents current state-of-the-art for Arabic sentiment analysis, the performance can further improve when integrating surface, syntactic and semantic features.

While most of the models described above focused on developing Arabic specific opinion mining models, several researchers explored translating Arabic text to English and then using state-of-the-art English sentiment classification systems as the works in [364, 365, 366, 367] or considered the outputs of different sentiment classification models for the same sentence in different languages for improved performance such as in [368]. [369] evaluated

state-of-the-art English sentiment system, SentiStrength [84] by applying it directly on 11 different Arabic texts and claimed to achieve an average F1 of 68%. Similarly, [370] compared two available online tools for sentiment analysis using a collected dataset of 1000 Facebook comments and tweets annotated for sentiment. They compared the performance of SentiStrength and SocialMention and claimed that SentiStrength performed better.

In summary, we present in Tables 2.4, 2.5 and 2.6 a summary of the opinion models discussed sampled from the different approaches described. The different opinion models categories presented in the above sections vary in their complexity in terms of how much information they use to infer sentiment from given text, resulting in different degrees of success. This also affects the complexity of training and using model. For instance, while lexicon-based approaches are simple and not highly accurate, they are very fast and light in terms of software requirements, which enables faster responses and model update, and easier integration into mobile applications [37]. On the other hand, while feature engineering approaches proved more successful, they require excessive efforts and time to create immense and sparse feature matrices, with many features possibly not necessarily existing in new texts. This imposes time limitations, especially when building new models or updating existing ones. These challenges related to language modeling and sparsity caused by Arabic complex morphology are addressed with deep learning models. While no consensus is reached in terms of the order of n-grams that should be used for feature-based approaches, deep learning models overcome this challenge by using word embeddings and having a complete embedded sentence representation. Moreover, deep learning models allow encapsulating semantic knowledge and sentence parsing structures in a condensed vector representation of the sentence. On the other hand, Deep learning models also require large scale training data, and excessive time and computing resources, with thousands or millions of parameters being learned for the task. Nevertheless, they are preferred over feature engineering approaches, due to the dense and compact input features (embeddings), which is one of the main reasons behind the success of these models. In Table 2.3, we summarize the challenges addressed (in bold) by the discussed categories of opinion mining models along with their respective drawbacks. We can observe that deep learning models address many Arabic opinion mining challenges but at the expense of creating large scale annotated corpora and of having enough computing power to learn the parameters of the network.

In general, developing opinion mining algorithms and models that are highly accurate has been the goal of several researchers, but, it is also very important to be able to integrate these algorithms and models into real world applications. Opinion mining applications are discussed next in section 2.5 highlighting the usage of opinion mining in several sectors.

OMA Technique	Addressed Challenges	Drawbacks
Lexicon Based Approaches	Limited Availability of large-scale annotated Arabic corpora for sentiment: No training needed. Language sparsity: Sentence represented by sentiment scores.	Requires external large scale lexical resources. Context is not taken into consideration. Accuracy depends on quality and size of lexicon.
Supervised Approaches Using Surface Features	Sentence representation: n-gram representation of a sentence. Context modeling: using a high order of n-grams to include context.	Increased sparsity with the increase of order of n-grams. No unique solution for the order of n-grams since it is corpus dependent.
Supervised Approaches Using Syntactic Features	Ambiguity of language: through lemmatization for example. Sentence parsing: understanding word relations.	Not sufficient on their own. Limited availability of NLP tools for the different Arabic dialects.
Supervised Approaches Using Semantic Features	Sentence Sentiment Extraction: using sentiment lexicons. Semantic relations across words: using AWN for example.	Limited availability of large scale semantic/sentiment lexicons or dictionaries such as AWN.
Deep Learning Models	Language Modeling. Language Sparsity. Sentence Representation. Semantic Relations. Sentence Parsing.	Limited availability of large scale Arabic sentiment annotated corpora for learning accurate models. Limited size of Arabic sentiment treebank compared to English. Computationally expensive.

Table 2.3: Arabic Opinion Mining Models: addressed challenges and drawbacks.

Reference	Preprocessing	Features Used	Classifier	Sentiment Lexicon	Corpus Used for Testing	MSA?	DA?	Evaluation Metrics	Class Labels	Results
Azmi & Alzain [2014] [254]	Removal of all words except those with sentiment connotation	N-grams	None	Naïve Bayes	Comments extracted from an Online edition of Saudi Newspapers	Yes	Yes	Precision, F1, Accuracy	Pos, Neg, Neu, Mixed	86.5%, 84.5%, 82%
Abtulla et al. [2014] [99]	Dialect elimination, Stemming, Intensification detection, Negation handling	Sentiment lexicon scores	Rule based, Unsupervised	Corpus based sentiment lexicon	Maktoob Corpus [230], Twitter Corpus [83]	Yes	No	Accuracy	Pos, Neg, Neu	74.6% 70.2%
Abtulla et al. [2014] [215]	Tokenization, Elongation correction, Normalization of Arabic characters, Light Stemming, Stop words removal, Intensification detection, Negation handling	Sentiment lexicon scores	Rule based, Unsupervised	Corpus based sentiment lexicon	Maktoob Corpus [230], Twitter Corpus [83]	Yes	Yes	Accuracy	Pos, Neg, Neu	63.8% 70.1%
Al-Ayyoub et al. [2015] [105]	Removal of non-Arabic characters, Elongation correction, Spelling correction, Stop words removal, Stemming, Negation handling	Sentiment lexicon scores	Rule based, Unsupervised	Expand seed list and translation from English sentiment lexicons	Own manually annotated Arabic tweets for sentiment	Yes	Yes	Accuracy	Pos, Neg, Neu	86.9%
Duwairi et al. [2015] [216]	Tokenization, Stemming	Sentiment lexicon scores	Rule based, Unsupervised	Translation of English Sentiment lexicon and Expansion using Arabic Thesauri	Own manually annotated Arabic tweets for sentiment	Yes	Yes	Precision, Recall, Accuracy	Pos, Neg	70.0% 46.0% 46.0%
Oraby et al. [2013] [218]	Tokenization, Parse Tree, Negation Handling, Intensifiers handling, Conjunctions detection	Sentiment lexicon scores	Rule based calculation of sentiment score per review	Translated version of MPQA	Opinion Corpus for Arabic (OCA) [164]	Yes	No	Absolute Error (AE)	Rating score between 1 and 10	2.3
Aly & Atiya [2013] [165]	Tokenization	Surface Features: 1, 2 and 3 grams, TFIDF	SVM	None	LABR Unbalanced	Yes	No	Accuracy and F1	Polarity and Rating Classification	Polarity: 91.0%, 90.1% Rating: 50.3%, 49.1%
Al Smadi et al. [2016] [291, 292]	Tokenization, segmentation, stemming, POS tagging, punctuation and stop words removal, normalization	POS tags, Named Entities, n-grams(n between 1 and 3)	CRF	None	Articles from Al Arabiya and Al Jazeera	Yes	No	Accuracy	Pos, Neg, Neu	86.5%
Omar et al. [2013] [262]	Tokenization, Stop words removal, Stemming, Dialectal and slang words conversion to MSA	Surface Features: Unigrams, bigrams, TFIDF	Ensemble: stacking and logistic regression	None	Reviews from jeeran.com manually annotated	Yes	Yes	Macro F1	Subjectivity classification and Pos, Neg	97.5% 91.0%

Table 2.4: Summary of Arabic Opinion Mining models (1/3).

Reference	Preprocessing	Features Used	Classifier	Sentiment Lexicon	Corpus Used for Testing	MSA?	DA?	Evaluation Metrics	Class Labels	Results
Duwairi (2015) [267]	Tokenization, Stop removal, Negation handling, Emoticons conversion to words, Replacing dialectal terms by MSA, Stemming	Surface Features: bag of words with binary representation	NB	None	Arabic annotated Tweets using Crowdsourcing	Yes	Yes	Macro F1	Pos, Neg, Neu	87.6%
Abdul-Mageed et al. (2014) [280]	Tokenization, Lemmatization, POS Tagging, Replacing low frequency words by UNIQUE	Sentiment lexicon binary features, Detect if MSA or dialect, Genre, User ID	SVM	Manually annotated lexicon of adjectives	Manually labeled data extracted from 4 different sources on the web	Yes	Yes	Accuracy	Subjectivity Classification, Polarity Classification	72.5% to 95.8% 65.9% to 81.4%
Al Shboul et al. (2015) [236]	Tokenization, Stemming, Stop words removal	Bag of words	MNB	None	LABR	Yes	No	Average F1	Rating classification from 1 to 5	42.8%
El Nagggar et al. (2017) [319]	Tokenization, Normalization	unigrams, bigrams, rule-based sentiment score, Emotion labels, number of sentiment and emotion tokens, presence of negations, count of total number of tokens in a tweet	SVM with bagging	Emoji lexicon	Own annotated Tweets	Yes	Yes	Accuracy	Pos, Neg, Neu	90%
Mourad & Darwish (2013) [91]	Tokenization, Stemming, POS Tagging	Stem, Stem-POS, Bigram stems, count of POS tags, Counts of stems belonging to strong or weak subjectivity classes, MSA or DA, Tweet specific features (presence of hashtags, user mentions, url, retweet), Presence of elongation, Presence of emoticons, Usage of decorating characters, Polarity from sentiment lexicon	NB	Combination of different lexicons	Manually annotated tweets	Yes	Yes	F1	Subjectivity Classification (subj/obj), Polarity Classification (pos, neg)	52.8%/71.0% 76.5%/66.4%
Ibrahim et al. (2015) [107]	Tokenization	Binary sentiment feature, TF, Polar word position, Negation, Intensifiers, question marks, suppletion and wishful expressions, POS tags, Bigrams, extraction of conflicting bigrams polarity	SVM	ArSeLEX and idioms for Egyptian dialect	Manually annotated dataset of tweets and reviews	Yes	Yes	F1	Pos, Neg	95.8%

Table 2.5: Summary of Arabic Opinion Mining Models (continued 2/3).

Reference	Preprocessing	Features Used	Classifier	Sentiment Lexicon	Corpus Used for Testing	MSA?	DA?	Evaluation Metrics	Class Labels	Results
Hamdi et al. (2018) [187]	Tokenization, Stemming, stop words removal, and oversampling	n-grams (n from 1 to 3), TFIDF	MNB; Bernoulli Naïve Bayes (BNB); Logistic Regression (LR); SVM	Mannually annotated lexicon	Mannually annotated corpus	Yes	Yes	Accuracy and F1	Pos, Neg, Neu	Up to 95%
Al Sallab et al. (2017) [31]	Tokenization, Lemmatization, Elongation adjustment, Replacement of URLs, hashtags and user mentions, Normalization	Word Embeddings, Sentiment Embeddings, Syntactic Parse Trees	Recursive Auto Encoder, Softmax layer	ArSenL	ATB, QALB, RR Tweets	Yes	Yes	Accuracy, F1	Pos, Neg	Per corpus: 86.5%/84.9% 79.2%/75.5% 6.9%/68.9%
Badaro et al. (2014) [36]	Tokenization, Lemmatization, Stop words removal	Three sentiment scores per sentence	SVM	ArSenL	ATB	Yes	No	Average F1	Subjectivity Classification, Polarity Classification, Pos, Neg	72.3% 64.5%
Gonzalez et al. (2017) [362]	Tokenization	Out-domain Word Embeddings (Wikipedia), In-domain word embeddings (SemEval Data), One hot encoding sentiment sequence vector	Stacking of CRNN, MLP, BLSTM. Softmax layer	NRC	SemEval 2017 Task 4-A	Yes	Yes	Recall, F1, Accuracy	Pos, Neg, Neu	47.8% 46.7% 50.8%
El Beltagy et al. (2017) [321, 371]	Removal of URLs and Diacritics, Elongation adjustment, Replacement of positive and negative emoticons by love and anger	Word embeddings obtained by running Word2Vec on 4 million tweets dataset, Bag of words, Overall sentiment of tweet predicted by a proprietary system, Features related to count and position of positive and negative terms	Ensemble Classification using CNN, MLP and LR	Proprietary lexicon	SemEval 2017 Task 4-B	Yes	Yes	Recall, F1, Accuracy	Pos, Neg	76.8% 76.7% 77.0%
Rushdi-Saleh et al. (2011) [164]	Tokenization, Stop words removal, Stemming, Filtering by token length	Surface Features: trigrams, TFIDF	SVM	None	OCA	Yes	No	Precision, Recall, Accuracy	Pos, Neg	87.4% 95.2% 90.6%

Table 2.6: Summary of Arabic Opinion Mining Models (continued 3/3).

2.5 Applications

After reviewing the different components of an opinion mining system that includes Arabic NLP tools, Arabic lexical resources and classification models, we present next applications to opinion mining with specific focus on recommender systems.

2.5.1 Opinion Mining Applications

Sentiment analysis applications evolved from being isolated applications that analyze sentences for subjectivity, to becoming vital entities in key sectors such as politics, healthcare, marketing, finance, services and education. Applications that have sentiment analysis at their core are emerging on continuous basis and are targeting all the above-mentioned sectors. However, only few ones rely on Arabic sentiment analysis. In what follows we present an overview of the most relevant work on Arabic sentiment analysis in each sector.

Politics Opinions that are shared on social media and blogging sites present valuable information that can be used for political purposes; to alert political leaders about potential problems or threats, to get a sense of how much a certain policy is being perceived by the public, to calculate a popularity index that can be used for elections [372], to know the public emotional status (angry, disgusted and happy) and many other usages. Determining the opinion holder [19] helps in developing such systems. Most recently, the authors of [373] implemented a system that, given a political figure, tracks the corresponding opinions presented on the web and presents a summarized report of that figure to monitor their political standing. Other relevant applications include: [374] focused on the effect of sentiment during the 2012 presidential elections in Egypt, [375, 376] proposed an automated tool to determine the political orientation of an Arabic article or comment and [377] analyzed the users' statuses on "Facebook" posts during the "Arabic Spring" era in Tunisia to get insights on the users' behavior. Alsmearat et al. in [378] were not able to find a correlation between the gender of a writer and the presence of opinion in Arabic text. In [379], Abu-Jbara et al. utilized Arabic sentiment analysis to detect subgroups in an online political debate. Given the opinions of discussants in a debate, discussants would belong to the same subgroup if they shared the same opinion about the same targets or topics.

Healthcare Many people share their health related data and experience on well-known blogs and on social media. People discuss their health issues, symptoms, diagnosis results, medication given, and their experiences when visiting the health care centers. It is many times very crucial to patients to know the experiences of other patients and consequently to take decisions of which health care center to visit or which medication to choose. This subject is

discussed to great detail in a book by Khan et al. [380]. In [350], the authors analyze the sentiments on health services from data collected on Arabic twitter.

Marketing Since social media and the web in general are being extensively used as a platform of customer interaction, sentiment analysis have taken marketing to a whole new level. Companies recognized the importance of sentiment analysis in branding their products and consequently invested heavily in recommender systems and social/sentiment analysis tools. Opinion mining can improve the quality of recommendation of recommender systems that are only based on user-item ranking matrix such as those described in [32, 33, 34]. In [230, 234, 381], the authors analyzed reviews and comments collected from Yahoo!-Maktoob, an Arabic social networking website. They extracted different aspects important for marketers such as the reviews' length, the numbers of likes/dislikes, the polarity distribution and the languages used. Wang et al. in [382] developed a social data analytics (SDA) tool that run on top of the IBM BigInsights platform. SDA allows identifying user characteristics like gender, locations, names, and hobbies; develop comprehensive user profiles across messages and sources; associate profiles with expressions of sentiment, buzz, intent, and ownership around brands, products, and companies. It enables data analysts with little knowledge in information extraction and sentiment analysis to get results from social data quickly. The authors added support for Arabic by developing a sentiment analysis model using Arabic tweets extracted about three different topics EGYPTIAN TELECO, Egyptian Government and Saudi Employment. Furthermore, Hathlian and Hafezs in [383] employed sentiment analysis on Arabic tweets in order to predict whether people are interested or not in a certain product or defined subject.

The interest in providing sentiment analysis for marketing purposes lead to the establishment of many companies that provide tools for Arabic sentiment analysis, such as Repustate⁵ and LexisNexis.⁶

Finance Sentiment Analysis is being used as a major factor in making financial decisions given the insights it gives on the subject matter under analysis [384]. Take for instance the insights sentiment analysis gives on the stock price movement [385]. Sentiment analysis is also used in measuring the mood of a given investor or the overall investing public, either Bullish or Bearish.⁷ In [386], the authors studied the impact of the Islamic holy month of Ramadan on Islamic Middle Eastern markets, where it was shown that there is always a positive increase in the stock market which can be attributed to the positive investor mood and sentiment during this month. Also in [387], the

⁵<https://www.repustate.com/sentiment-analysis/>

⁶<https://www.lexisnexis.com/en-us/news-company-research/default.page>

⁷www.investorwords.com

authors suggested a trading strategy with Mubasher products, a leading stock analysis software provider in the Gulf region, using sentiment analysis from tweets. Al-Rubaiee et al. in [388, 389] utilized Saudi Twitter posts to predict Saudi stock market. The authors studied the relationship between opinion from social media and the Saudi market index in order to help foreign investors gain an insight into the opinions of Saudi investors. Alkubaisi et al. in [390] proposed a sentiment analysis model on Arabic tweets for stock market price prediction for Al-Marai dairy company. Alshahrani et al. in [391] investigated the impact of sentiment in Arabic Tweets on Saudi Stock Market indicators. They manually annotated a set of 114K tweets for sentiment and they evaluated different correlation metrics with the price change in Saudi Stock Market. In [392], a granger causality is found between the amount of sentiment tweets and the stock market price changes in the Arab world.

Services Sector Ahmed et al. in [393] present an application of sentiment analysis using natural language toolkit (NLTK) for measuring customer service representative (CSR) productivity in real estate call centers. The study describes in details the decisions made, step by step, in building an Arabic system for evaluating and measuring productivity. The system includes transcription method, feature extraction, training process and analysis. In [394], Rahamatallah et al. presented a sentiment analysis system prototype to specifically analyze customer reviews of Sudanese Telecommunication products. In [395, 396], Arabic sentiment analysis was used to measure customer satisfaction towards Telecommunication companies in Saudi Arabia. Similarly, Najadat et al. in [397] also employed Arabic opinion mining to measure customer satisfaction towards Jordanian Telecommunication companies.

Education Arabic sentiment analysis has been used to evaluate students' satisfaction and experience at university by analyzing students' tweets [398]. Arabic opinion mining was also used to understand the opinion of students towards colleges [399]. El-Halees in [400] presented a system to track changes of opinions expressed by Arab students about their courses in order to improve course evaluation in the future. Add to that many studies are currently being conducted to evaluate the experience of students who enroll in or take online MOOCs.

Sentiment models can also be integrated into recommender systems models to improve the accuracy of recommendation. We present next a detailed overview of typical recommender systems models.

2.5.2 Recommender Systems

Recommender systems are being utilized in different scenarios such as recommending social events based on the user geographical information [401], helping users in selecting their travel packages [402], recommending web pages [403], solving patent maintenance related problems [404] and last but not least for recommending collaborators in scientific research across different domains [405]. Recommender systems methods can be thought of as a matrix completion problem where the goal is to find the missing ratings of a user-item rating matrix. Recommender systems algorithms are mainly divided into collaborative filtering techniques: user-based and item-based ones, content-based recommender systems, hybrid recommender systems and preference-based recommender systems. The latest trends in recommender systems are providing justifications of the recommended items [406, 407] and the use of machine learning specifically deep learning such as MLP, RNN or CNN for nonlinear transformation and sequence modeling [408].

Collaborative Filtering Techniques Sarwar et al. presents in [409] a technique that makes use of collaborative filtering. This technique assumes a list of m users $U = \{u_1, u_2, u_3, \dots, u_m\}$ and a list of n items $I = \{i_1, i_2, i_3, \dots, i_n\}$. Each user u , has rated a list of items noted by I_{ui} . The purpose of this technique is to predict the ratings of unrated items by a given user and recommend the Top-N items.

Two approaches for collaborative filtering, which are mainly user-based and item-based, can be distinguished as follows: user-based collaborative filtering utilizes the similarity computed between the active user and all other users; item-based collaborative filtering makes use of the similarity available between two items. The similarity measures rely on the ratings available for two queried items. For instance, two users who provided close ratings to the same set of items will most likely have a similarity measure close to 1, whereas two users who have different ratings for the same set of items are more likely to have similarity measure close to 0. Computing the similarity measure can be done using the Pearson correlation coefficient, the cosine similarity or the adjusted cosine similarity as described in [410]. An example of Pearson correlation coefficient is given in 2.1. For users u and v , I_{uv} represents the set of items rated by both users u and v , r_{ui} the rating of user u to item i and \bar{r}_u the average of ratings provided by user u .

$$sim(u, v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_v)^2}} \quad (2.1)$$

For each of the two above approaches, user-based and item-based collaborative filtering, two major algorithms are described: memory-based collaborative filtering algorithms and model-based collaborative filtering

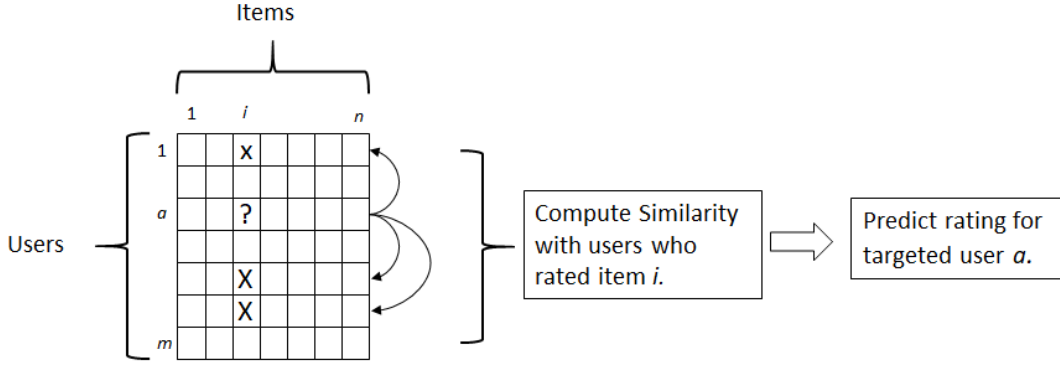


Figure 2.1: Collaborative Filtering process using user-based approach.

algorithms. Memory-based algorithms exploit the entire user-item rating matrix to perform a prediction. Statistical techniques are employed to find the nearest neighbors for a user if one considers a user-based approach or for an item if one considers an item-based approach. The predicted rating r is given in equation 2.2 based on user-based approach where v belongs to the user in the neighborhood of the active user u :

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N(u)} sim(u, v) * (r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u)} |sim(u, v)|} \quad (2.2)$$

In order to normalize the result, the mean rating of the user v is subtracted from v 's rating for item I and is divided by the sum of the absolute value of the computed similarities in order to make sure that the predicted rating fall within the ratings range, as for example between 1 and 5. Item-based collaborative filtering was proved to have a lower mean absolute error 2.3 compared to user-based collaborative filtering [409]. For each prediction pair $\langle p_i, q_i \rangle$, p_i being the predicted value and q_i the correct value available in the training data, the absolute error is computed as $|p_i - q_i|$. For each prediction pair $\langle p_i, q_i \rangle$, p_i being the predicted value and q_i the correct value available in the training data, the absolute error is computed as $|p_i - q_i|$.

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \quad (2.3)$$

The MAE is then evaluated by examining N ratings-prediction pairs, and computing the average error as shown in the equation below. An example of user-based collaborative filtering process can be summarized in Fig. 2.1.

On the other hand, model-based algorithms suggest an item recommendation by first developing a model of user ratings using different machine learning techniques such as Bayesian network, clustering methods and

rule-based approaches [410] and thus the collaborative filtering is treated as a classification problem.

In order to improve the accuracy of typical collaborative filtering techniques, several methods have been proposed [27]. Liu et al. in [411] proposed a new similarity measure to improve the rating prediction using collaborative filtering, such as aggregating external ratings [412], or applying matrix factorization and its variations [413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423]. Recently, a compressive sensing framework has been proposed for recommender systems using collaborative filtering by Gogna and Majumdar [424].

Content-based Recommender Systems In [26], the authors describe content based recommendation methods which are inspired from information retrieval and information filtering research. The content-based recommender system looks at the content of a certain item depending on its type and tries to analyze the commonalities among the items that the user has highly rated. Then, based on the analysis, the system should detect items with degree of similarity to the users preferences. Typical applications for content-based recommender systems are the ones that deal with items with textual information, such as documents, web sites or movies descriptions. The issues with this approach is that it cannot generalize to multiple applications since the content is different from one type of item to another. Thus, the need for an approach that can be generalized to multiple product types.

Hybrid Models Hybrid models were developed to address the limitations of collaborative filtering techniques and content-based methods. The limitation of collaborative filtering is the problem of cold start user and items, i.e. users who did not provide enough ratings and items which were not rated yet as described in [425, 426]. As for content-based recommender systems, extracting content is not feasible for all types of items.

Adomavicius and Tuzhilin in [26] and Claypool et al. in [427] describe four ways to merge collaborative filtering and content based models. The first one works by combining the separate recommenders ratings using a linear combination or a voting scheme which basically selects the recommendation that is seen better in terms of quality and more consistent with past users ratings. As for the second method, it adds the content-based characteristics to collaborative models which can help in overcoming the sparsity problem since the model is not only relying on ratings but also on item profiles for our prediction. The third way is to add collaborative characteristics to content-based models where latent factors are introduced to describe the user preferences. The fourth way is to develop a single unifying recommendation model based on content-based and collaborative characteristics using probabilistic approaches such as rule-based classifier or Bayesian regression

models.

Preference-based Recommender Systems A newly introduced approach for recommender systems is the preference-based one that instead of relying on item ratings provided by the users, it identifies abstract features and relations based on the user profile. For instance, based on the profession of the user, his/her location, his/her gender and other preferences, the preference-based recommender system constructs abstract relationships that are relatively more general and can capture a larger set of people with similar preferences extracted from their specific profiles. The authors in [428] introduce a preference-based recommender system for the conference recommendation problem that recommends conference sessions for a new set of users based on user profiles and conference themes constructed using nonnegative matrix factorization. The advantage of the proposed technique is that it does not require item ratings but instead it relies on user behavior and sessions attendance for this specific case. Other possible types of matrix projection that are used to learn abstract relationship about users or items based on tangible data are Singular Value Decomposition, Principal Component Analysis and Vector Quantification. The preference-based model was adopted by several ecommerce companies such as Amazon.com and Netflix [429].

Recommender Systems Challenges As the amount of information is increasing tremendously on a daily basis and as the number of E-commerce users is also increasing, challenges for existing recommender systems are being more crucial to tackle. Scalability of recommender systems is currently a major point to achieve in order to accommodate for the increasing number of users/items [430]. Moreover, since recommender systems are heavily based on users interventions on the web and their opinions, privacy issues should not be violated by recommender systems and this makes it challenging for systems that rely for instance, on the cache of a web-browser [431]. Collection of data is sometimes hard to achieve. Hence, implicit and user-friendly ways are required to avoid having sparse data and simplifying the retrieval of ratings from users. Typical techniques for collecting user input on items are performed when the user is signing up for the first time on the website or through a survey. Last but not least, evaluation metrics are being discussed more recently since sometimes a low root-mean-square-error system does not guarantee a good quality recommendation. In fact, users are becoming more curious in knowing why these specific items were recommended and not others. In the same context, research in recommender systems is now considering recommending new items not only based on preference of the users but also on providing them with diversified items that they did not explore before as described in [432].

Integrating Sentiment Analysis with Recommender Systems Some work has been done for including sentiment analysis in recommender system models. Karampiperis et al. in [433] proposed a recommender system for educational content that takes into consideration the opinion of the user. They claim that including the actual opinion of the user improve the quality of the recommendation. They study how user comments can help in providing a recommendation when a rating is missing. They propose a simplistic approach for finding the sentiment of the reviews by using the counts of predefined positive and negative terms and computing a rating score based on the counts. They test their system on MERLOT platform. While Karampiperis et al. in [433] focused on improving the accuracy of recommender systems for educational content, Ganu et al. in [29, 434] proposed an ad-hoc and regression based measures, which take into account the opinion of the user, for recommending restaurants. Their results show that using textual information results in better general or personalized review score predictions than those derived from the numerical star ratings given by the users. They claim that their techniques allow them to make a more fine-grained recommendation. They evaluate their approach using restaurant reviews extracted from NY Citysearch website. They make use of a multivariate regression which learns weights or importance to be associated with the different textual information. Last but not least, Pappas and Popescu-Belis in [435] propose a sentiment aware nearest neighbor model for multimedia recommendations over TED Talks. They include in their model labeled and unlabeled user-comments and claim that their method outperforms several baseline approaches in terms of accuracy by 25%. They make use of a dictionary based sentiment classification methods and map the sentiment label to a rating score in order to incorporate it in their model.

After going through an overview of the literature concerning the different components of the dissertation, we cover next in the upcoming chapters the achieved contributions in terms of resources, analytics and applications.

Chapter 3

Lexical Resource Expansion: Heuristic Techniques

In this chapter, we describe the different sentiment and emotion lexical resources that were developed using heuristic approaches. The lexicons are ArSenL [36], EmoWordNet [43] and ArSEL [45].

Before getting into the details of the creation of each resource, we briefly describe the list of existing resources that were used in the different approaches.

3.1 Background on Existing Resources

3.1.1 Arabic WordNet (AWN)

AWN [20] contains a list of Arabic synsets, where each synset represents a group of synonym words that share the same sense meaning. AWN's XML format file consists of three parts: the first part is the synset representative terms, which are also known as synset IDs, the second part is the set of synonym terms or expressions that represent the same sense. The third part is the links that reflect the semantic relations across the synsets such as hyponyms, hypernyms and related_to. An example from AWN is shown in Table 3.1. AWN 2.0 contains 10,456 synsets and around 19,000 Arabic expressions.

3.1.2 English WordNet (EWN)

EWN [436] has around 120,000 synsets that include synset terms, glosses and extended glosses. A synset contains a group of terms or expressions that share the same meaning. A gloss give a definition of the synset and an extended gloss provides examples to reflect the specific sense usage of a given synset. EWN is one of the most used resources for English NLP. Several synset ID-linked versions of EWN have been released (2.0, 2.1 and 3.0). The synset ID is a unique identifier

Synset ID / Synset Representative Term	Words in the Same Synset	Semantic Links to Other Synsets
202418477 / zaAra.v1AR (زار)	\$aAhada.1 (شاهد) ra>aY.3 (رأى) zaAra.1 (زار)	zaA}ir_n2AR (زائر) Semantic Link: related_to
201938649 / saAra.v1AR (سار)	\$aAraka_fiy_mawokib.1 (شارك في موكب) ma\$aY.6 (مشى) saAra.2 (سار) saAra_fiy_mawokib.1 (سار في موكب) taqad~ama.11 (تقدم) xaraja_fiy_masiyrap.1 (خرج في مسيرة)	ma\$aY.v1AR (مشى) Semantic Link: hypernym

Table 3.1: Examples of AWN synsets.

for a synset in EWN. EWN includes a dictionary augmented with lexical relations (synonymy, antonymy, etc.) and part-of-speech (POS) tags. An example of EWN synset is represented in Table 3.2. EWN 3.0 contains 117,659 synsets and around 155,000 English terms. By excluding POS, EWN has 147,306 terms out of which 64,188 are multiword terms (~44%).

Synset ID	POS	Synset Representative	Synset Terms	Gloss	Extended Gloss	Semantic Link to Other Synsets
07547805	n	hostility#3	hostility#3 enmity#2 ill_will#3	the feeling of a hostile person	“he could no longer contain his hostility”	hate#1 Semantic Link: hyponym

Table 3.2: Detailed example for an EWN synset.

3.1.3 English SentiWordNet (ESWN)

English SentiWordNet (ESWN)[144, 9] is a large-scale English Sentiment lexicon that provides for each synset in EWN 3.0 three sentiment scores whose sum is equal to 1: Pos, Neg, and Obj. In brief, ESWN is the English WordNet augmented with sentiment scores. The sentiment scores were generated using a semi-supervised approach. An example of ESWN is shown in Table 3.3.

3.1.4 Standard Arabic Morphological Analyzer (SAMA)

SAMA 3.1 [437] is a commonly used morphological analyzer for Arabic. Each lemma has a POS tag and English gloss as shown in the example in Table 3.4.

Synset ID	POS	Synset Representative	Synset Terms	Gloss	Extended Gloss	Pos. Score	Neg. Score	Neu. Score
07547805	n	hostility#3	hostility#3 enmity#2 ill_will#3	the feeling of a hostile person	“he could no longer contain his hostility”	0.00	0.125	0.875

Table 3.3: Detailed example for an ESWN synset.

The gloss in SAMA is a set of English words that reflect the meaning of the Arabic lemma. The analyzer produces for a given word all of the possible lemma readings out of context. SAMA includes 32 granular POS tags and the majority of POS tags are nouns, proper nouns, verbs and adjectives. SAMA English gloss includes 10,110 multiword expressions out of 28,020 terms (36%, excluding POS).

SAMA Lemma	POS	English Gloss
majAEap (مجاعة)	n	famine; starvation

Table 3.4: Example of an entry in SAMA.

3.2 ArSenL: A Heuristic-based Link Prediction Approach

In this section, a heuristic based link prediction approach is presented for creating a large scale Arabic Sentiment Lexicon (ArSenL). The approach consists of combining the outcomes of mapping AWN 2.0 to ESWN 3.0 through offset sense mapping and linking SAMA lemmas to ESWN synsets through gloss terms to synset terms matching. The work was published in [36] and received so far more than 95 citations. The development of ArSenL fueled the research on Arabic opinion mining and accelerated the work of my colleagues on state-of-the-art deep learning models for Arabic opinion mining as described in AROMA [31]. Moreover, as described in [38, 44], ArSenL helped our OMA team to win in competitions related to sentiment analysis specifically in SemEval 2017 [205] and SemEval 2018 [361]. Last but not least, ArSenL helped in developing efficient and accurate sentiment classification models for mobile application development [37] described in section 5.1.

3.2.1 Methodology

The approach consists of linking the Arabic lexical resources to the English resources using heuristics. The flow of the approach is shown in Fig. 3.1. Starting with the two Arabic resources SAMA and AWN, we would like to link them to ESWN. Mapping AWN to ESWN consists of two major steps: first

mapping AWN 2.0 synsets to EWN 3.0 synsets and thus to ESWN 3.0 using WordNet sense-map files, and second updating AWN Arabic lemmas to the Linguistic Data Consortium (LDC) format which is the same as SAMA lemma transliteration form. Standardizing AWN lemma transliterations involved exact matching between the two resources AWN and SAMA, applying some modifications, summarized in Table 3.5, to AWN lemma forms to match SAMA lemma forms or backing off using the SAMA morphological analyzer on AWN terms and selecting the lemma with the lowest edit distance.

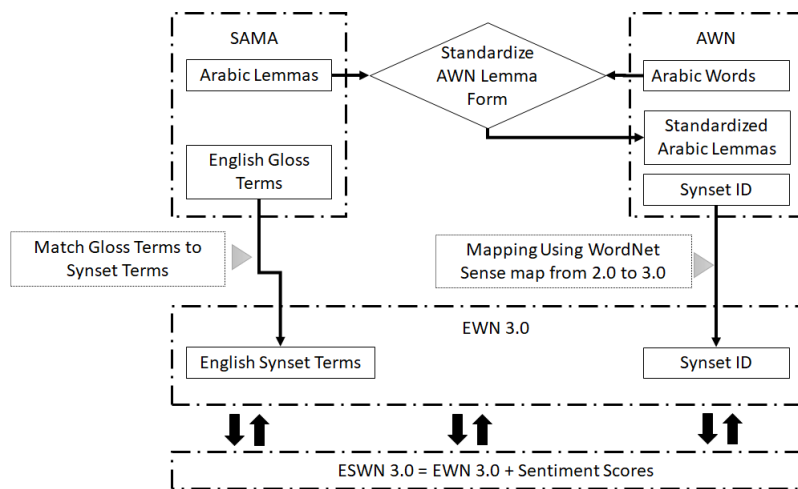


Figure 3.1: Mapping SAMA and AWN to ESWN.

AWN	After Modification	Example
aA	A	kifaAH → kifAH (struggle)
If lemma is a verb and it ends with “a”	Remove “a”	\$aAEa → \$AE (circulate)
If lemma ends with K	Replace K by iy	mADK → mADiy (past)

Table 3.5: Summary of modifications performed to AWN lemmas in order to match them to SAMA lemma LDC form.

We denote this sub-approach by the Arabic WordNet approach. Since AWN was manually developed, the link between the resources is assumed to be 100% accurate. Although the precision of this method is very high, the recall of this method is low since only sentiment scores for 4,507 lemmas were retrieved. The second sub-approach denoted by the English-based approach consists of mapping SAMA to ESWN through matching the gloss terms of SAMA to ESWN: a SAMA lemma is linked to an ESWN synset if any of the its gloss terms match any of the ESWN synset terms while making sure that pos tags match and that the links generated correspond to the ones with the largest overlap between the gloss terms and the synset terms. An example of how this approach works is shown in Figure 3.2. This approach requires little manual effort but it is more

prone to noise: i.e., it has a much higher recall but a lower precision. Through this approach 28,540 lemmas ($\sim 76\%$ of SAMA) were mapped to ESWN. The validation of ArSenL-Eng was performed (a) automatically by using ArSenL-AWN and (b) manually by randomly validating 400 distinct lemmas. For the automated part, we check for each common lemma between the two lexicons if the sentiment scores match. A total of 3,833 lemmas (out of 4,507) from ArSenL-AWN were matched in ArSenL-Eng. Thus, we can inspect that the precision of the remaining scores is of 85%. For the manual validation, we check if the meaning of the SAMA lemma corresponds to the one in ESWN. 70% of the 400 randomly selected lemmas were accurately mapped to ESWN. The main issue of the remaining 30% is the unavailability of enough glosses per SAMA lemma, which makes it harder to heuristically map to EWN synsets.

ArSenL is the result of combining the two generated lexicons ArSenL-AWN and ArSenL-Eng as described above respectively. The union of the two lexicons consisted of combining the two resources and adding a field in the lexicon to distinguish the original source of the entry. For instance, an entry from the first approach, i.e. ArSenL-AWN, will have an AWN offset while an entry in ArSenL-Eng will have the same field set to N.A (Not Available). Furthermore, manual correction was performed to ArSenL-AWN for the AWN lemmas that were mapped to SAMA using minimum edit distance. The gold version of the union lexicon includes 28,780 lemmas with the corresponding number of 157,969 synsets. Moreover, ArSenL is available publicly through www.oma-project.com. Table 3.6 summarizes the sizes of the lexica used. It is important to note that these numbers are obtained by excluding the POS tags and including for verbs the vowels that represent the present tense of the verb.

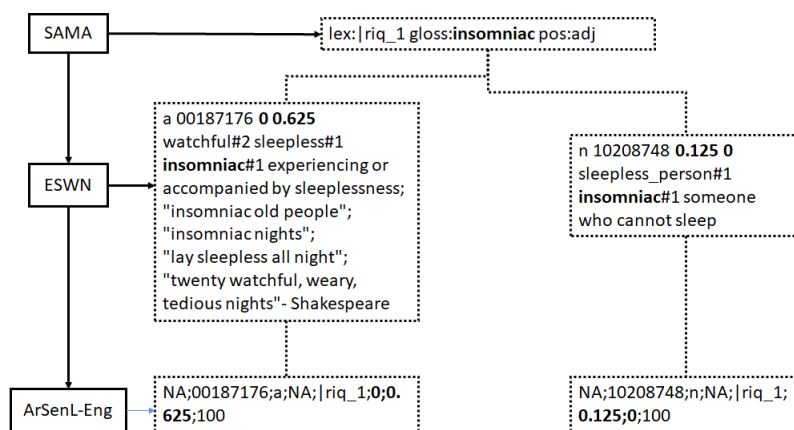


Figure 3.2: Steps to map SAMA lemma to ESWN synset.

Lexicon	Number of Lemmas	Number of Related Synsets
Automatic Process		
ArSenL-AWN	4,507	7,326
ArSenL-Eng	28,540	150,700
ArSenL-Union	28,812	158,026
Manual Correction		
ArSenL-AWN	4,492	7,269
ArSenL-Union	28,780	157,969

Table 3.6: Sizes of created sentiment lexica.

3.2.2 Evaluation of ArSenL

We conduct an extrinsic evaluation to compare the different versions of ArSenL on the task of subjectivity and sentiment analysis (SSA). We also compare the performance of ArSenL to the one of SIFAAT lexicon [77].

Experimental Settings We perform our experiments on the same corpus used by AbdulMageed et al. [77]. The corpus consists of 400 documents from the Penn Arabic Treebank (part 1 version 3) [46] that are gold segmented and lemmatized. The sentences are tagged as objective, subjective-positive, subjective-negative and subjective-neutral. We use nonlinear SVM implementation in MATLAB, with the radial basis function (RBF) kernel, to evaluate the different lexicons in the context of SSA. The classification model is developed in two steps. In the first step, the kernel parameters (kernel width γ and regularization parameter C) are selected, and in the second step the classification model is developed and evaluated based on the selected parameters. To decide on the choice of RBF kernel parameters, we use the first 80% of the dataset to tune the kernel parameters to the values that produce the best F1-score using 5-fold cross-validation. The resulting parameters are then used to develop and evaluate the SVM model using 5-fold cross-validation on the whole dataset.

Two experiments were conducted to evaluate the impact of the different lexicons on opinion mining. The first experiment considers subjectivity classification where sentences are classified as either subjective or objective. In this experiment, the SVM kernel parameters were tuned to maximize the F1-score for predicting subjective sentences. The second experiment considers sentiment classification, where only subjective sentences are classified as either positive or negative. Subjective-neutral sentences are ignored. In this experiment, the classifiers parameters are tuned to maximize the average F1-score of positive and negative labels. We report the performance measures of the individual classes, as well as their average.

For baseline comparison, the majority class is chosen in each of the experiments, where all sentences are assigned to the majority class. For subjective versus objective baseline classification, all sentences were classified as

subjective since the majority (55.1%) of the sentences were subjective. To further emphasize the importance of detecting subjectivity, we chose the F1-score for subjective as baseline. For positive versus negative baseline classification, all sentences were classified as negative since the majority (58.4%) of the dataset was annotated as negative. The resulting baseline performance measures are captured in Table 3.7, and serve as basis for comparison with our developed models. For the subjective versus objective the baseline F1-score is 71.1%, and for positive versus negative, the baseline F1-score is averaged as 36.9%.

		Baseline	ArSenL			SIFAAT
			AWN	Eng	Union	
Coverage %		NA	56.6	88.8	89.9	32.1
Subjective	F1	71.1	71.2	72.1	72.3	66
	Precision	55.1	58.1	58.5	58.3	61.5
	Recall	100	92	93.9	95.1	71.4
Positive	F1	0	52.9	59.7	61.6	55.4
	Precision	0	44.7	55	55.2	51.8
	Recall	0	64.8	65.6	70.1	60.2
Negative	F1	73.7	67	70.7	75.6	67.6
	Precision	58.4	46.4	60.6	61	59.4
	Recall	100	53.9	62.4	64.5	59.2
Average F1(Pos/Neg)		36.9	53.9	62.4	64.5	59.2

Table 3.7: Results of extrinsic evaluation. Numbers that are highlighted reflect the best performances obtained by the lexicons, without considering the baseline.

Features We train the SVM classifier using sentence vectors consisting of three numerical features that reflect the sentiments expressed in each sentence, namely positivity, negativity and objectivity. The value of each feature is calculated by matching the lemmas in each sentence to each of the lexicons separately: ArSenL-AWN, ArSenL-Eng, ArSenL-Union and SIFAAT. The corresponding scores are then accumulated and normalized by the length of the sentence. We remove all stop words in the process based on [438]. For words that occur in the lexicon multiple times, the average sentiment score is used. It is worth noting that the choice of aggregation for the different scores and the choice of nonlinear SVM were concluded after a set of experiments. In this regards, we conducted a suite of experiments to evaluate the impact of using: (a) linear versus Gaussian nonlinear SVM kernels, (b) normalization based on sentence length, (c) normalization using z-score versus not, and (d) using the confidence score from the lexicons. Our best results across the different configurations reflected the best results with the nonlinear Gaussian RBF kernels, with sentence length-based normalization and without confidence weighting.

Results Three evaluations were conducted to compare the performances of the developed sentiment lexicons. The results of the experiments are shown in

Table 3.7. First, we evaluate the coverage of the different lexicons. We define coverage as the percentage of lemmas (excluding stop words) covered by each lexicon. ArSenL-AWN and SIFAAT have lower coverage than the ArSenL-Eng lexicon. The union lexicon has the highest coverage. This is normally due to the larger number of lemmas included in the English and union lexicons, as shown in Table 3.6. In subjectivity classification, ArSenL lexicons perform better than the majority baseline and outperform SIFAAT in terms of F1-score. Overall, the developed ArSenL-Union gives the best performance among all lexicons. The only exception of better performance for SIFAAT for subjectivity is in terms of precision, which is associated with a much lower recall resulting in an F1-score that is lower than that of ArSenLs. Similarly, sentiment classification experiment reveals that ArSenL lexicons produce results that are consistently better than SIFAAT and the majority baseline. The ArSenL-Union lexicon outperforms all lexicons in all measures without exceptions.

In summary, it can be observed that the English-based lexicon produces result that is superior to the AWN-based lexicon. Combining both resources, through the union, allows further improvement in SSA performance. It is also worth noting that the English and union lexicons consistently outperform SIFAAT despite the fact that the latter was manually derived from the same corpus we are using for evaluation. We close by showing examples of ArSenL in Table 3.8. The lemmas are in their Buckwalter [61] format for easier integration in any NLP task. The word NA stands for Not Applicable. In the case where AWN Offset is NA and AWN lemma is NA, this means that the entry is retrieved from ArSenL-Eng. Otherwise, the entries are from ArSenL-AWN. The additions to the lemmas such as `_v1AR` , `_n1AR`, `_1` or `_2` can be dropped when data processing is performed. They were kept for easier retrieval in the original sources (AWN and SAMA). We added the English Gloss field for easier understanding of the Arabic word in the table. Moreover, it can be seen that only positive and negative scores are reported in the lexicon since the objective score can be easily derived by subtracting the sum of positive and negative scores from 1.

SAMA Lemma	POS tag	Positive Score	Confidence %	Negative Score	AWN Offset	ESWN Offset	AWN Lemma	English Gloss
Aap-1	n	0	100	0	NA	4151581	NA	screen
\$ATir-1	a	0.75	33	0	NA	1335458	NA	smart;bright
\$Alhidap-1	n	0	50	0	NA	5820620	NA	proof
\$Al-u-1	v	0	50	0	NA	792921	NA	lift
\$Amix-1	a	0.75	33	0	NA	1285136	NA	superior
danA'ap-1	n	0.222	33	0.778	NA	4730580	NA	inferiority
Hazin-a-1	v	0	50	0.5	NA	1797347	NA	sorrow
sAxin-1	a	0.75	50	0.125	NA	811421	NA	hot
faraH-1	n	0.5	33	0.25	NA	7527352	NA	joy
Hasan-1	a	0.625	100	0	NA	64787	NA	good
{izodahar-1	v	0.125	100	0	200300610	310386	<izodahara_v1AR	flourish
>a\$oEar-1	v	0	100	0	200844607	873682	>a\$oEara_v1AR	notify
>aHobaT-1	v	0.125	100	0.5	201766276	1819147	>aHobaTa_v1AR	discourage
nahAr-2	n	0	100	0	114279405	15136453	nahaAr_n1AR	day
najAH-2	n	0.625	100	0	100059106	64504	najaAH_n1AR	success
niykol-1	n	0	100	0	113808178	14646610	naykl_n1AR	nickle
tabAyum-1	n	0.25	100	0.625	104540432	4748836	tabaAyum_n1AR	difference
tasA'al-1	v	0.375	100	0	200705236	729378	tasaA'ala_v1AR	wonder
\$ariyf-2	a	1	67	0	NA	1983162	NA	respectable
\$ariyr-1	n	0	33	0.75	NA	5144663	NA	evil

Table 3.8: Samples of ArSenL showing entries originating from ArSenL-Eng and ArSenL-AWN.

3.3 EmoWordNet: Automatic Expansion of Emotion Lexicon Using English WordNet

The following work was published in [43]. The lexicon helps in extending ArSenL to include emotion scores as described in the next section 3.4.

3.3.1 Methodology

In this section, we describe the approach we followed in order to expand DepecheMood and build EmoWordNet. DepecheMood consists of 37,771 lemmas along with their corresponding POS tags where each entry is appended with scores for 8 emotion labels: afraid, amused, angry, annoyed, don't care, happy, inspired and sad. Three variations of score representations exist for DepecheMood. We select to expand the DepecheMood variation with normalized scores since this variation performed best according to the presented results in [130].

In Fig. 3.3, we show an overview of the steps followed to expand DepecheMood.

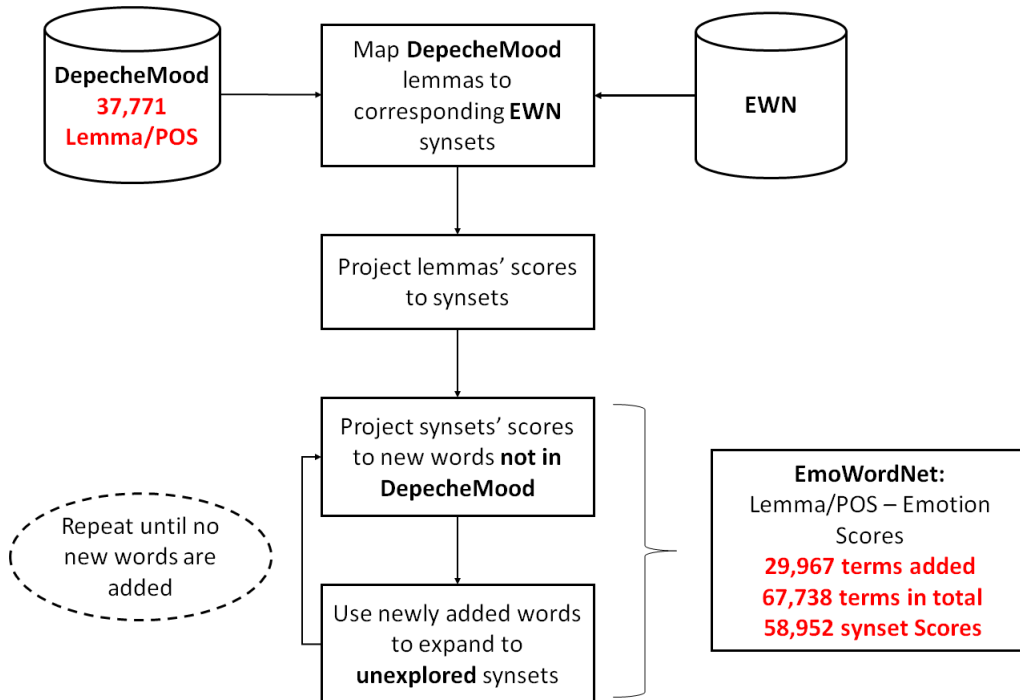


Figure 3.3: Overview of DepecheMood expansion approach.

Step 1: EWN synsets that include lemmas of DepecheMood were retrieved. A score was then computed for each retrieved synset, s . Let S denotes the set

of all such synsets. Two cases might appear: either the retrieved synset included only one lemma from DepecheMood, in this case the synset was assigned the same score of the lemma, or, the synset included multiple lemmas that exist in DepecheMood, in this case the synset’s score was the average of the scores of its corresponding lemmas.

Step 2: A synset, s , includes two set of terms: T , terms that are **in** DepecheMood, and \bar{T} , terms **not in** DepecheMood. Using the synonymy semantic relation in EWN, and based on the concept that synonym words would likely share the same emotion scores, we assigned the synset’s scores to its corresponding terms \bar{T} . Again, a term t in \bar{T} might appear in one or multiple synsets from S . Hence, the score assigned to t would be either the one of its corresponding synset or the average of the scores of its corresponding synsets that belong to S .

Step 3: after performing step 2, new synsets might be explored. Terms in \bar{T} might also appear in synsets \bar{s} that do not belong to S . \bar{s} would get the score of its corresponding terms. Step 2 and 3 were repeated until no new terms or synsets were 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 mostly preserves the emotion orientation and does not require manual validation as described by [122].

As a walking example of the steps described above, let us consider the DepecheMood term “bonding” having noun as POS tag. “bonding” can be found in three different EWN noun synsets 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 EWN synset so there are no more iterations.

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 present a simple extrinsic evaluation of EmoWordNet similar to the one performed for DepecheMood.

3.3.2 Evaluation of EmoWordNet

In this subsection, we evaluate the effectiveness of EmoWordNet in emotion recognition task from text. We evaluate regression as well as classification of emotions in unsupervised settings using similar techniques used for evaluating

DepecheMood.

3.3.2.1 Dataset & Coverage

We utilized the dataset provided publicly by SemEval 2007 task on Affective text [132]. 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 [120] while those in EmoWordNet, as well as DepecheMood, follow the ones provided by Rappler Mood Meter. We considered the same emotion mapping assumptions presented in the work of [130]: Fear \rightarrow Afraid, Anger \rightarrow Angry, Joy \rightarrow Happy, Sadness \rightarrow Sad and Surprise \rightarrow Inspired. Disgust was not aligned with any emotion in EmoWordNet and hence was discarded as also assumed in [130]. One important aspect of the extrinsic evaluation was checking the coverage of EmoWordNet against SemEval dataset. In order to compute coverage, we performed lemmatization of the news headlines using WordNet lemmatizer available through Python NLTK package. We excluded all words with POS tags different than noun, verb, adjective and adverb. EmoWordNet achieved a coverage of 68.6% while DepecheMood had a coverage of 67.1%. An increase in coverage was expected but since the size of the dataset is relatively small, the increase was only around 1.5%.

In terms of headline coverage, only one headline (“Toshiba Portege R400”) was left without any emotion scores when using both EmoWordNet and DepecheMood since none of its terms were found in any of the two lexicons.

3.3.2.2 Regression and Classification Results

We followed an approach similar to the one presented for evaluating DepecheMood. For preprocessing, we first lemmatized the headlines using WordNet lemmatizer available in Python NLTK package. We also accounted for multi-word terms that were solely available in EmoWordNet by looking at n-grams (up to n=3) after lemmatization. We then removed all terms that did not belong to any of the four POS tags: noun, verb, adjective and adverbs. For features computation, we considered two variations: the sum and the average of the emotion scores for the five emotion labels that overlapped between EmoWordNet and SemEval dataset. Using average turned out to perform better than when using sum for both lexicons. As stated in [130] paper, ‘Disgust’ emotion was excluded since there was no corresponding mapping in EmoWordNet/DepecheMood. The first evaluation consisted of measuring

Pearson Correlation between the scores computed using the lexicons and those provided in SemEval. The results are reported in Table 3.9. We could see that the results are relatively close to each other: EmoWordNet slightly outperformed DepecheMood for the five different emotions. It was expected to have close results given that the coverage of EmoWordNet is very close to DepecheMood. Given the slight improvement, we expect EmoWordNet to perform much better on larger datasets.

For the classification task, we first transformed the numerical emotion scores of the headlines to a binary representation. We applied min-max normalization on the computed emotion scores per headline, and then assigned a ‘1’ for the emotion label with score greater than ‘0.5’, and a ‘0’ otherwise. We used F1 measure for evaluation. Results are shown in Table 3.10. More significant improvement was observed in the classification task compared to the regression task when using EmoWordNet.

Emotion	EmoWordNet	DepecheMood
Fear	0.59	0.54
Anger	0.42	0.38
Joy	0.33	0.21
Sadness	0.43	0.40
Surprise	0.51	0.47
Average	0.46	0.40

Table 3.9: Pearson Correlation values between predicted and golden scores.

Emotion	EmoWordNet	DepecheMood
Fear	0.45	0.32
Anger	0.17	0.00
Joy	0.48	0.16
Sadness	0.46	0.30
Surprise	0.43	0.40
Average	0.40	0.24

Table 3.10: F1-Measure results for emotion classification.

3.3.2.3 Results Analysis

In this section, we present some quantitative and qualitative analyses of the results. For quantitative analysis, we checked first whether the count of terms in a headline is correlated with having a correct emotion classification. Overall, the length of headlines was varying between 2 and 15 terms. Headlines with length between 5 and 10 terms were mostly correctly classified. Hence, one can conclude that having a headline with couple of terms only may not allow the system to clearly decide on the emotion label and having headlines with many terms may cause the system to over predict emotions. In addition to headline length,

we checked whether POS tags are correlated with correct or erroneous emotion predictions. Given that the dataset consists of news headlines, the “noun” POS tag was the most frequent in both correctly classified headlines and misclassified ones.

For qualitative analysis, we analyze few correctly classified headlines and few other misclassified ones. We show in Table 3.11 few examples of correctly classified headlines and in Table 3.12 other examples of misclassified headlines. By looking at the misclassified examples, we observe that the golden annotation tend to be sometimes conflicting such as the second and the fifth examples in Table 3.12 where we have joy and sadness as assigned emotions for the two headlines. An explanation for having conflicting emotions for the same headline is that the annotators reflected their personal point of view of the information conveyed by the headline. Hence, some people were happy to read the headline others were sad. In order to incorporate such challenging aspect of emotion recognition from text, more sophisticated emotion recognition models need to be considered and tested.

Headline	Emotions
Hackers attack root servers	Anger; Fear
Subway collapse caught on camera	Fear; Sadness
Action games improve eyesight	Joy; Surprise
Study finds gritty air raises heart disease risk in older women	Fear; Sadness; Surprise
Wizardry at Harvard: physicists move light	Surprise

Table 3.11: Examples of correctly classified headlines.

Headline	Gold	Predicted
A film star in Kampala, conjuring aminos ghost	Fear; Surprise	Anger; Joy; Sadness
Damaged Japanese whaling ship may resume hunting off Antarctica	Joy; Sadness	Anger; Fear; Surprise
Apple revs up Mac attacks on Vista	Surprise	Anger; Fear; Joy; Sadness
Serbia rejects United Nation’s Kosovo plan	Anger; Sadness; Surprise	Fear; Joy
Taliban leader killed in airstrike	Joy; Sadness	Anger; Fear; Surprise

Table 3.12: Examples of misclassified headlines.

We presented EmoWordNet, a large scale emotion lexicon, consisting of around 67K EWN words and 58K EWN synsets annotated with 8 emotion scores. EmoWordNet is automatically constructed by applying a semantic expansion approach using EWN and DepecheMood. When utilized for emotion recognition, EmoWordNet outperformed existing emotion lexicons and had a better lexical coverage.

3.4 ArSEL: A Large Scale Arabic Sentiment and Emotion Lexicon

In this section we present the work for developing ArSEL an extended version of ArSenL that includes emotion scores assigned to ArSenL entries via EmoWordNet described in section 3.3. The work was published in [45].

3.4.1 Methodology

ArSEL development relies on the developed resource EmoWordNet [43]. Fig. 3.4 summarizes the overall methodology for constructing ArSEL.

After creating EmoWordNet as described in subsection 3.3.1, 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 lemma-pos pairs, i.e., 94.71% of ArSenL lemmas. We summarize the lexicon sizes per lemma in Table 3.13.

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

Table 3.13: Lexicons coverage.

As a walking example of the steps described above, we added to the steps shown in Fig. 3.4 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” **لحام** and hence the Arabic lemma will be assigned the emotion scores of the synset.

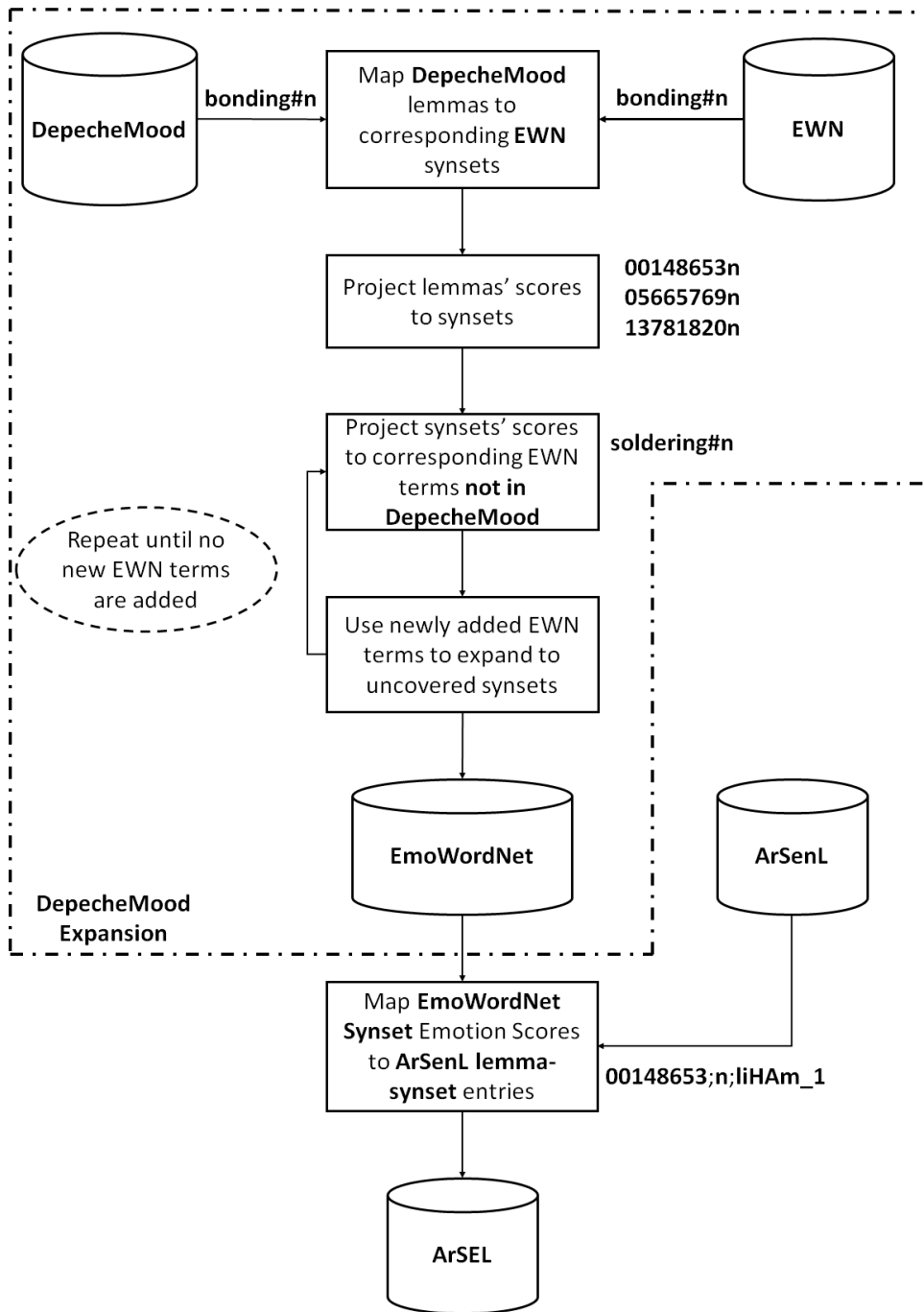


Figure 3.4: Overview of ArSEL construction methodology.

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. We also show some sample lemmas of ArSEL along their corresponding 8 emotion scores in Table 3.14. We have picked samples that

should be emotionally charged to check if the emotions represented by the lemma have the highest scores.

3.4.2 Evaluation Using SemEval 2007 Dataset

Since ArSEL is generated based on ArSenL, the intrinsic evaluation results of ArSenL described in [36] 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.

3.4.2.1 About the Dataset

We utilize SemEval 2007 Affective Task dataset [132]. 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 [120] 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 [130] and summarized in Table 3.16. Disgust emotion label in SemEval is not aligned with any emotion in EmoWordNet and hence is discarded as also assumed in [130]. 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 3.15.

3.4.2.2 Experiment Setup

We perform the following preprocessing steps in order to proceed with the evaluation. We utilize MADAMIRA [57] in order to perform lemmatization for the translated dataset. The output of MADAMIRA is a list of lemmas in Buckwalter transliteration [439] 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

Lemma#POS	English Gloss	Afraid	Amused	Angry	Amoyed	Don't Care	Happy	Inspired	Sad
xawof#n 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 شخص	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	0.111311906	0.06151737	0.096234049
kuway~is#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 3.14: Sample of ArSEL Arabic lemmas with emotion scores.

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

Table 3.15: News' Headlines' examples to show differences between Google translations and human translations.

scores and the scores provided in SemEval taking into consideration the mapping of emotion labels as represented in Table 3.16. 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 Fig. 3.5.

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

Table 3.16: Mapping between SemEval and ArSEL emotion labels.

3.4.2.3 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 ر.٤٠٠”) 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 3.17, 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 3.15. 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 from using ArSEL on the translated version of the same dataset. We compute the average of the two resulting scores and use it to

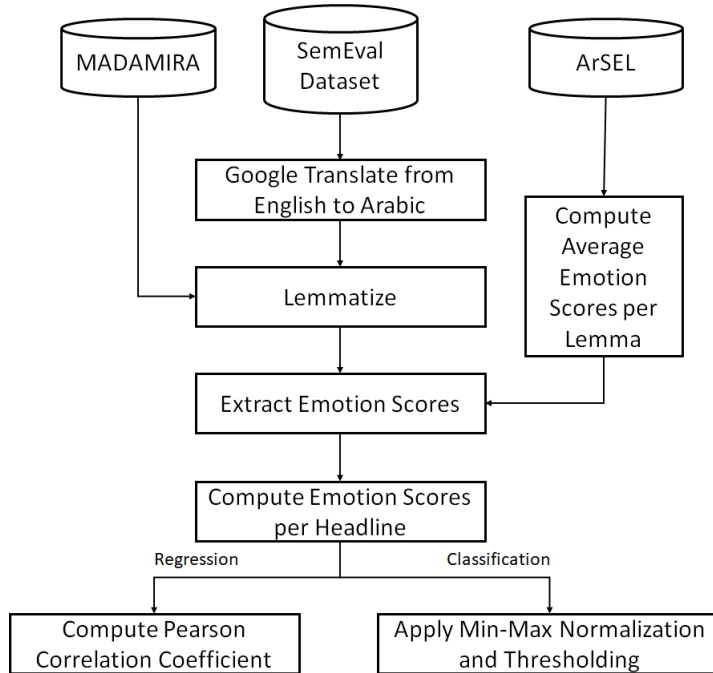


Figure 3.5: Overview of ArSEL evaluation steps.

perform regression and classification. We report the regression results in Table 3.17 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 3.18, we also compare F1 measure achieved by using ArSEL and EmoWordNet on translated SemEval and original one respectively. We also test the performance of combining the output of the two lexicons based on the parallel dataset shown under combined column in Table 3.18. 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 3.20, we show some examples of news' headlines that were correctly classified and in Table 3.21, 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.

Emotion	EmoWordNet	ArSEL	Combined
Fear	0.59	0.44	0.53
Anger	0.42	0.34	0.37
Joy	0.33	0.26	0.35
Sadness	0.43	0.31	0.41
Surprise	0.51	0.1	0.14
Average	0.46	0.29	0.36

Table 3.17: 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 3.18: 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.

3.4.3 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 3.4.2.2.

3.4.3.1 About the Data

In SemEval 2017, a task was created for Arabic Twitter sentiment analysis [205]. Several teams participated and the winning teams were NileTMRG [321] and OMAM [346, 440]. 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

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 3.20: 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 summi 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 3.21: Examples of misclassified News' headlines from SemEval 2007.

have only access to the training and the development sets. In total, there are 2,871 tweets. In Table 3.19, we show the distribution of emotions across the tweets. The frequencies of the emotions are very close to each other with “Sadness”

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

Table 3.19: Distribution of emotion labels across the tweets.

being the most frequent in the dataset. We follow the same experiment setup described in section , 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 (خااa

3.4.3.2 Regression and Classification Results

We follow the same approach described in section related to SemEval 2007 dataset to perform regression and classification with the modifications described in the section about the SemEval dataset of 2018. 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 3.22 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 3.23. 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 achieved when utilizing more sophisticated regression and classification techniques.

We also show examples of correctly classified tweets in Table 3.24, whereas in Table 3.25, we present examples of misclassified tweets. By analyzing some of the misclassification examples we can see that several tweets are in dialectal

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

Table 3.22: 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 3.23: Classification F1-score results on SemEval 2018 Arabic tweets dataset.

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 3.25: 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.

We presented 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 is 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.

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 هالفترة عباره عن دموع وحرقت قلب؟	Why is this period full of fears 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 يا لجمهرة كلك لطف وحيابة 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 3.24: Examples of correctly classified Arabic tweets from SemEval 2018.

Arabic Tweet and English Translation	True Emotions	Predicted Emotions
كنت اكثر انساها بنخاف بس صارلي كم سنه كثير جريته حتى بحضر افلام رعب لخالي 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 بيلاش 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 كانوا ييشوفو ان اللي عيونهم ملونه نذير شوم تحسي عينه فيها غدر وشر That's why Arabs thought that people with colored eyes are evil	fear	joy; sadness
شيء ما يقوم بإشغال قتل 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 3.25: Examples of misclassified Arabic tweets from SemEval 2018.

Chapter 4

Lexical Resource Expansion: Machine–Learning Techniques

We present in this chapter a machine learning formulation for our link prediction problem. A summary of this work is currently under review to be submitted to ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP).

Link prediction methods typically involve a two-step process. In the first step, a similarity is assessed between the two targets, and in the second step the similarity score is compared to a threshold for deciding on the presence or absence of a link. In this work, we propose different approaches to match the terms in the two lexicons. Depending on the matching approach, different similarity measures are also considered: Jaccard similarity measure and Cosine Similarity measure. The explored features along with their respective challenges are summarized as follows:

- **Direct comparisons between base SAMA English gloss terms and EWN synset terms:** Here, the similarity assessment compares SAMA English terms to EWN English synset terms using Jaccard similarity.
 - While EWN synset terms are lemmatized, SAMA English terms are not lemmatized and include special characters such as “() + /” and quotation marks.
 - SAMA English translation terms are limited and do not cover the complete list of possible translations for each term. Similarly, EWN synset terms tend to be very sense specific. This limitation can sometimes result in having similarity scores of 0 between SAMA English translation terms and EWN synset terms although the terms might be synonyms. As a result, a high recall cannot be guaranteed. For instance, the SAMA lemma *kuroh* (كره), has as English translations ‘hatred’ and ‘loathing’ and thus will result with a

Jaccard similarity of 0 when comparing it with its respective EWN synset ‘dislike’. EWN considers ‘loathing’ as a hate coupled with disgust and ‘hatred’ as an intense dislike.

- **Comparison between expanded SAMA terms with machine translation (MT) tables and base EWN terms:** Here, the similarity assessment compares an expanded set of SAMA English translations to EWN terms using a variation of the Jaccard similarity measure that takes into consideration probability scores showing the confidence of a certain MT table entry as shown in section 4.3.4. The expansion of SAMA English terms is based on publicly available machine translation (MT) tables derived from aligned Arabic-English parallel corpora [441].

- While MT tables help in expanding the coverage of SAMA English translations, it also introduces the challenge of increased noisy Arabic to English translations, i.e., incorrect Arabic to English translations. The increase in noisy translations may lead to an increase in predicting inaccurate links between SAMA lemmas and EWN synsets, hence, a lower precision but with a higher recall. Hence, it is required to integrate a measure of the reliability in the translated terms such as the use of probability scores that accompany Arabic to English entries in MT tables into the similarity measure. It is also important to note that the choice of the parallel corpora has an impact on the performance of this method. The selected MT table [441] consists of translated Arabic News on different topics. This increases the chance of matching to EWN synset terms. For instance, using scientific parallel corpora may result in different accuracy results.

- **Comparison between word embeddings of SAMA and EWN English terms:** Here, the similarity assessment compares the semantic similarity between SAMA English translation terms and EWN synset terms using vector dot product. Pretrained word embeddings model are generated based on raw textual data and a vector representation is assigned for each single word term. Using word embeddings help in improving the recall of our link prediction model.

- The new challenge here is that EWN synset terms and SAMA English translations include several multiword expressions that cannot be directly assigned a vector representation.

The methods are evaluated for their strengths and limitations and we also evaluate the performance of using ensemble models that utilize the best of these features.

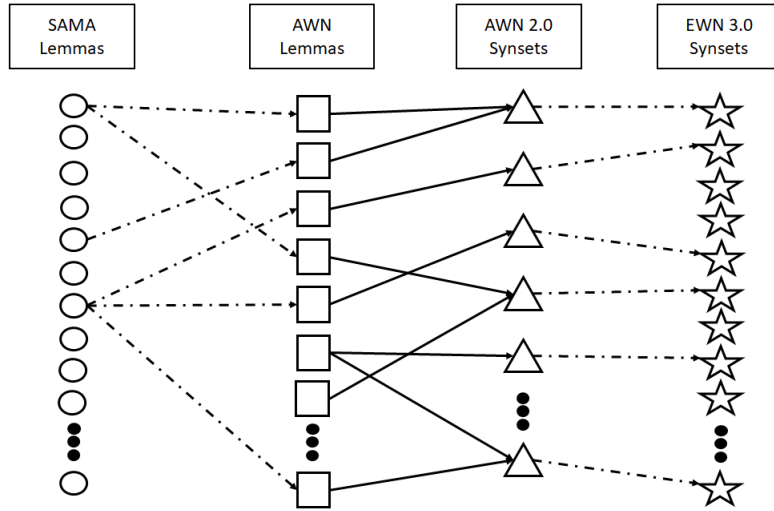


Figure 4.1: Overview graph of the training data development for AWN expansion. The dotted arrows represent the mapping described in section 4.1.

4.1 Developing Gold Dataset

By excluding proper nouns and particles POS tags, the objective is to map around 35,000 SAMA lemmas to EWN.

The objective of the gold set of links is to form a basis for developing and testing the proposed models that can automatically link SAMA lemmas to EWN synsets. To achieve this gold set, two steps are followed. In the first step, we identify common terms between AWN and SAMA. In the second step, we identify the AWN-EWN standards for linking terms. The overall mapping process is shown in Fig. 4.1, where the circle represents a SAMA lemma, the square an AWN lemma, the triangle an AWN synset, and the star an EWN synset. It is true that the process of developing the gold dataset overlaps with the work described for creating ArSenL-AWN in subsection 3.2.1. However, it is important to note that, unlike ArSenL-AWN, adjectives are included in the development of the gold dataset and the proper nouns are excluded. Moreover, more manual efforts were put into the development of the gold dataset. Therefore, we describe in details how the gold dataset is formed.

4.1.1 Matching AWN Lemmas to SAMA Lemmas

To identify the common terms between SAMA and AWN, AWN needed some preprocessing and cleanup since its current format includes multiple inconsistencies and does not conform to well established formats for Arabic NLP tools. On the other hand, SAMA lemmas are compliant with the Linguistic Data Consortium (LDC) format. To map the AWN words to SAMA

lemmas, we needed to find the correct LDC lemma representation of AWN lemmas. For example, the word “Jordan” (أردن) is transliterated in six different ways in AWN: *AlArdn*, *AAlArdn*, *Al>rdn*, *>rdn*, *Ardn* and *Al>urodun*.

Hence, the first task of matching AWN lemmas to SAMA LDC standard forms is not a trivial task. Additionally and as can be seen in Fig. 4.1, some SAMA lemmas are not mapped to any AWN lemmas since SAMA is around 4 times bigger than AWN in terms of lemma counts. Furthermore, AWN includes multiword terms while SAMA consists of single word lemmas. We exclude multiword terms in this process. Moreover, given the inconsistencies in AWN lemmas, one SAMA lemma can map to multiple AWN lemmas. On the other side, an AWN synset includes multiple AWN lemmas, hence multiple AWN lemmas can link to one AWN synset. Moreover, one AWN lemma can have multiple senses and thus, can appear in different AWN synsets.

AWN consists of 11,322 unique lemma forms/POS tag (including inconsistencies and inaccuracies) while SAMA consists of 40,691 lemma/POS tag. AWN has four POS tags (noun, verb, adverb, adjective) while SAMA has 32 tags. The mapping process between AWN lemmas and SAMA lemmas consists of two steps. To achieve the AWN-SAMA matching, we first propose to generate simple rules that can identify mismatches in the formats for the same terms. These rules provide an initial list of directly matched terms. We then manually check for the lemmas that were not detected in the first step, and process those terms manually by first generating additional possible variations of the terms.

(a) Set Derived by Automatic Matching

The automatic approach consists of two steps:

- i- Matching AWN words without any modification to SAMA lemmas while at the same time checking that the part of speech tag matches in the two resources. In this step we were able to map 3,186 AWN words.
- ii- The remaining unmatched AWN words, 8,136 words, from step 1 were modified based on a set of rules identified by comparing the two resources. The rules are reported in Table 3.5. In addition, we normalized all the beginning Hamza to “A” in both AWN and SAMA. In this step, we were able to map additional 2,589 AWN words.

(b) Manual Matching

Lemmas that were not matched in part (a) were manually checked using Arabic Lexeme-base Morphological Generation and Analysis System (ALMOR), which uses SAMA as its database. Given an AWN word,

ALMOR [1, 2] generates several out of context possible matches. An example of the output generated by ALMOR for the AWN lemma saAwaY (ساوى) is shown in Table 4.1. We are interested in the lex (lemma), gloss and pos fields. By manually comparing the corresponding EWN terms of a given AWN lemma and the SAMA English gloss terms, the correct SAMA lemma is selected. During the manual process, two issues in AWN lemma writing were encountered:

- i- Inaccuracies in terms of spelling of the lemma in AWN: for example, the verb (اضحك/make laugh) is written as >aHaka instead of >aDHaka, the verb (ادرك/realize) is written as >a*roka instead of >adoraka, and the noun (صيغة/Formula) is written as Sigap instead of Siygap.
- ii- Inconsistencies in lemma spelling for the same word in AWN. This inconsistency leads to inaccurate count of lemmas in AWN since the same lemma is transliterated in different forms in AWN but mapped to one unique form in SAMA as the example of Jordan (أردن). Another example includes, (اخضاع/Subjection), which is written in four different forms in AWN: <xoDAE, <xoDaAE, <xDAE and <ixDAE.

The results were manually checked and the accurate SAMA lemma was selected to map it to the AWN word by comparing SAMA glosses and WordNet synset terms of the corresponding AWN lemma. Some of AWN words did not generate any correct output in ALMOR as for example: natorata, natoraja, galowana and SaAlaba. In this manual step, we were able to map additional 3,652 AWN words.

(c) Resulting Common AWN-SAMA Set

To summarize the mapping between SAMA and AWN, AWN has 11,322 unique single words lemma forms to be matched to SAMA. We were able to map 9,427 of them. The 9,427 in AWN corresponded to 7,711 SAMA lemmas. 528 of the 7,711 SAMA lemmas have proper noun as POS tag. However, since proper nouns do not carry semantic information, we decided not to include these 528 proper noun lemmas in the gold data set. Thus, we end up with 7,183 unique lemmas/POS in SAMA mapped to AWN. To ensure the accuracy of the matching process, we validated 300 random lemmas and we got a 100% accurate matching.

Input to ALMOR	saAwaY
Output	diac:sAwaY lex:sAwaY.1 bw:+sAwaY/PV+(null)/PVSUFF_SUBJ:3MS gloss:+settle;be.equivalent;equalize+he/it.[verb] pos:verb prc3:0 prc2:0 prc1:0 prc0:0 per:3 asp:p vox:a mod:i gen:m num:s stt:na cas:na enc0:0 rat:na source:lex stem:sAwaY stemcat:PV_0

Table 4.1: Example of the output generated by ALMOR [1, 2] for a given Arabic lemma.

4.1.2 Mapping AWN to EWN

For the second step of linking AWN to EWN, we first recognize that AWN version is 2.0, while EWN is currently in version 3.0. Hence, there is a need to identify the proper mapping between AWN 2.0 synsets and EWN 3.0 synsets. In our previous work [36], we developed a large-scale Arabic sentiment lexicon (ArSenL). Part of ArSenL’s development process included linking AWN to EWN. However, the linking process was limited to AWN synset representative terms and to nouns and verbs only. To map AWN 2.0 synsets to their corresponding EWN 3.0 synsets, we use a set of files that provides a variety of IDs to map the different versions of AWN and EWN. The files are called EWN sense map files¹. Each AWN 2.0 synset can be linked to one EWN 3.0 synset. Out of the 10,456 synset IDs that exist in AWN, we were able to map 10,434 synsets to EWN 3.0. We could not achieve 100% mapping between the two resources due to missing mappings in the sense map files.

4.1.3 Resulting Gold Set

The result of the previous two steps is a set of accurate links between common lemmas in SAMA/AWN and EWN 3.0 synsets. An example of the whole mapping is shown in Table 4.2. The resulting details of the matched sets are reported in Table 4.3 along with the corresponding POS tags. Since we obtained only 2 lemmas with adverb as POS tag, we discarded them for the gold links. In Table 4.4, we show the total number of links in the gold data for the three POS tags: nouns, verbs and adjectives. We split these links into training set (90%) and testing set (10%) such that there are no common lemmas between the two sets.

4.2 Challenges in Shallow Link Prediction

Given the links in the gold data, we would like to predict accurately new links between EWN and SAMA terms not in AWN. Hence, based on the linking between 7,183 Arabic lemmas and 7,683 EWN synsets, we would like to predict

¹<http://www.talp.upc.edu/index.php/technology/resources/multilingual-lexicons-and-machine-translation-resources/multilingual-lexicons/98-wordnet-mappings>

SAMA Lemma	POS	AWN Words	AWN 2.0 Synset ID	EWN 3.0 Synset ID	EWN Terms	Synset
\$a>on (شأن)	noun	\$a>n_n1AR	105343715	05670710	concern	
		\$a>n_n2				
		\$a>n_n2AR	105515753	05855004	thing	
		\$a>n_n5				
		\$a>n_n3	105344294	05671325	thing, matter, affair	
		\$a>on_1	113152313	13943968	thing	
		\$a>n_1	104877294	05169813	significance	
\$a<on_1	107753466	08252211	social gathering, social affair			
\$a>n_4	105478649	05814650	issue			

Table 4.2: Example of the mapping across the different resources.

POS	Count of AWN Lemmas	Count of SAMA Lemmas	Count of EWN Synsets
Noun	7,490	4,452	4,857
Verb	2,664	2,153	2,361
Adjective	1,022	576	463
Adverb	146	2	2
Total	11,322	7,183	7,683

Table 4.3: Count of lemmas mapped from AWN to SAMA by POS tag.

POS	Training		Testing		Total	
	Lemmas	Links	Lemmas	Links	Lemmas	Links
Noun	4,007	8,271	445	915	4,452	9,186
Verb	1,939	4,771	214	525	2,153	5,296
Adjective	519	658	57	67	576	725
Total	6,465	13,700	716	1,507	7,181	15,207

Table 4.4: Count of gold lemmas and links per POS tag.

accurate potential links between around 35K SAMA Arabic lemmas and 117K EWN synsets.

The prediction problem has several challenges:

- **Comprehensive coverage of links in gold set:** While the terms in AWN are already linked to some terms in EWN, the list of potentially correct links is not complete. As a result, the gold set of links is not comprehensive and does not capture all correct links. Using an automatic evaluation of the system becomes challenging as it may result in incorrectly labeling false positives and getting lower than expected precision. For example:

- The SAMA lemma xAlaf_v (خالف) with English SAMA gloss terms ‘contradict’; ‘conflict_with’; ‘go_against’ can be linked to the following EWN synset 02378851v with synset terms ‘go_against#3’ ‘buck#2’ and synset gloss and extended gloss: ‘resist’; ‘buck the trend’ but this

link is not present in the gold set.

- The SAMA lemma Hur~_a (حر) with gloss terms ‘free’; ‘independent’ can be linked to the following EWN synset 01065694a with synset terms ‘free#6’ and gloss and extended gloss ‘not held in servitude’; ‘after the Civil War he was a free man’. While the link is semantically correct, it is also not present in the gold set.
- The SAMA lemma laHom_n (لحم) with English definitions ‘meat’; ‘flesh’ can be linked to the following EWN synset 05268112n with synset terms ‘flesh#1’ and extended gloss ‘the soft tissue of the body of a vertebrate: mainly muscle tissue and fat’. This link is also not present in the gold set.

- **Limited SAMA Gloss Terms:** SAMA gloss terms do not include all possible English translations for a given Arabic lemma. Consequently, some links in the gold data cannot be retrieved through simple direct matching. For example:

- The SAMA lemma ra>os_n (رأس) with gloss terms ‘chief’; ‘head’; ‘leader’; ‘top’ should be mapped to the following EWN synset 05601198n with synset term ‘face#7’ and extended gloss ‘the part of an animal corresponding to the human face’. However, the term face is not present in SAMA gloss terms.
- The SAMA lemma muEaT~il_n (معطل) with gloss terms ‘blocker’; ‘jammer’ should be mapped to the following EWN synset 0744164n with synset terms ‘vacationist#1’ ‘vacationer#1’ and extended gloss ‘someone on vacation’; ‘someone who is devoting time to pleasure or relaxation rather than to work’.
- Based on AWN–EWN mappings, the SAMA lemma takal~us_n (تكلس) with gloss terms ‘calcareous degeneration’; ‘calcification’ should be mapped to the following EWN synset 05645199n with synset terms ‘mental_block#1’ ‘block#7’ with extended gloss ‘an inability to remember or think of something you normally can do’; ‘often caused by emotional tension’; ‘I knew his name perfectly well but I had a temporary block’. However, SAMA English gloss does not include the sense related to the brain function takal~us zihoniy~ (تَكَلُّسٌ ذِهْنِي).

- **Different Choices of Words in SAMA versus EWN to Represent the Same Meaning:** In some cases, SAMA English gloss terms can be semantically linked to EWN synset terms, but the terms used in one lexicon are different from the terms with the same meaning in the other lexicon.

Similarity measures based on exact term matching will fail to capture such links. For example:

- The SAMA lemma ra}iys_n (رئيس) with English gloss terms ‘chairman’; ‘head’; ‘president’ should be linked to the following EWN synset 09623038n with synset term ‘leader#1’ and extended gloss ‘a person who rules or guides or inspires others’.
 - The SAMA lemma kafAlap_n (كفالة) with English definitions ‘bail’; ‘collateral’; ‘deposit’ should be linked to the following EWN synset 06685456n with synset terms ‘warranty#1’ ‘warrantee#3’ ‘warrant#4’ ‘guarantee#1’ with extended gloss ‘a written assurance that some product or service will be provided or will meet certain specifications’.
 - The SAMA lemma nAl_v (نال) with English definitions ‘achieve’; ‘acquire’; ‘attain’; ‘be_achieved’; ‘be_acquired’; ‘be_attained’; ‘confer’; ‘grant’ should be linked to the following EWN synset 01100145v with synset terms ‘win#1’ and extended gloss ‘be the winner in a contest or competition’; ‘be victorious’; ‘He won the Gold Medal in skating’; ‘Our home team won’; ‘Win the game’.
- ***Multiword Terms in One Lexicon that Can Match to Single Terms in Another Lexicon:*** SAMA English gloss terms as well as EWN terms include many multiword expressions. Multiword terms create a challenge when it comes to direct matching as well as when it comes to having a vector representation. Splitting the multiword terms into single words and matching can help addressing this issue but at the expense of possibly lowering the overall precision of the system if splitting is not done appropriately. For example:
- The SAMA lemma EalAmap_n (علامة) with English definitions ‘mark’; ‘point’; ‘sign’ should be linked to the following EWN 07417851n with synset terms ‘watershed#3’ ‘turning_point#1’ ‘landmark#2’ and extended gloss ‘an event marking a unique or important historical change of course or one on which important developments depend’; ‘the agreement was a watershed in the history of both nations’.
 - The SAMA lemma muqAbil_n (مقابل) with English definitions ‘in_compensation_for’; ‘in_exchange_for’ should be linked to the following EWN 13291189n with synset terms ‘offset#2’ ‘counterbalance#3’ with extended gloss ‘a compensating equivalent’.
 - The SAMA lemma munotajaE_n (منتجع) with gloss terms ‘resort’; ‘vacation_place’ should be linked to the following EWN synset

08640739n with synset terms ‘vacation_spot#1’ ‘resort_area#1’ ‘playground#1’ and extended gloss ‘an area where many people go for recreation’.

- **Matching to Homographs or Homonyms:** although SAMA English gloss terms may match exactly to EWN synset terms, the predicted links may be inaccurate in terms of meaning representation. In other words, the Arabic lemma is not the appropriate translation of the expressed EWN sense. For example:

- The SAMA lemma \$Ahad_v (شاهد) with gloss terms ‘watch’; ‘observe’; ‘witness’, by direct matching, can be linked to multiple wrong EWN synsets such as the synset with terms ‘watch_over#1’ ‘watch#2’ ‘observe#7’ ‘keep_an_eye_on#1’ ‘follow#13’ and with gloss and extended gloss ‘follow with the eyes or the mind’; ‘Keep an eye on the baby, please!’; ‘The world is watching Sarajevo’; ‘She followed the men with the binoculars’. For this example, the correct SAMA lemma that should be linked to ‘watch_over’ EWN synset is rAqab (راقب). On the other hand, the gold EWN synsets that correspond to the lemma \$Ahad (شاهد) are: the first synset has as term ‘watch#1’ and as gloss and extended gloss ‘look attentively’; ‘watch a basketball game’ and the second synset has as synset terms ‘watch#3’ ‘view#3’ ‘take_in#6’ ‘see#7’ ‘catch#15’ and extended gloss ‘see or watch’; ‘view a show on television’; ‘This program will be seen all over the world’; ‘view an exhibition’; ‘Catch a show on Broadway’; ‘see a movie’.
- The SAMA lemma qi\$or_n (قشر) with SAMA English definitions ‘skin’; ‘shell’; ‘peel’; ‘scale’, by direct matching, can be linked to the following EWN synset with terms ‘shell#7’; ‘racing_shell#1’ and extended gloss ‘a very light narrow racing boat’. But, the Arabic lemma does not correspond to the meaning expressed by the synset.
- SAMA lemma qamoE_n (قمع) with gloss term ‘repression’ can be linked to the following synset by direct matching although the meaning expressed by the synset does not correspond to the Arabic lemma. The synset has as term ‘repression#2’ and as extended gloss ‘(psychiatry) the classical defense mechanism that protects you from impulses or ideas that would cause anxiety by preventing them from becoming conscious’. The correct lemma should be kabit (كبت).

4.3 Proposed Link Prediction Methods

The methods explored for solving the link prediction problem and achieving a highly accurate prediction of SAMA lemma–EWN synsets links can be grouped into four main categories:

- **Direct matching of SAMA and EWN lexical terms:** These approaches consist of using SAMA English translations and EWN synset terms to compute similarity measures. Pre-processing of the terms includes lemmatization, stemming, multiword splitting, and stop words removal.
- **Using EWN extended glosses:** SAMA English translations are matched against EWN extended glosses.
- **Using MT tables:** in order to expand the coverage of SAMA English translations, MT tables are utilized to compute similarity with EWN synset terms.
- **Using word embeddings:** word embeddings are utilized to assess semantic similarity between SAMA English translations and EWN synset terms.

We describe first the details on the training and testing data, the evaluation metrics, the NLP tools that are used in the same way across the different methods explored. We then describe in details the methods used in each of the above categories. After exploring the different sets of techniques, we propose a fusion model.

4.3.1 Training and Testing Sets

The gold data set described in subsection 4.1 is split into a training set ($\sim 90\%$) and a testing set ($\sim 10\%$). The split is performed based on SAMA lemmas, i.e., no Arabic lemma exists simultaneously in both training and testing sets. Moreover, we perform the split for each POS tag separately. Detailed numbers about the count of SAMA lemmas, EWN synsets and SAMA-EWN links for each split are shown in Table 4.4 above.

For evaluation metrics, we report on precision, recall, and F1. We explore two variations for the three metrics. The first variation (V1) involves measuring the overall precision, recall and F1, while the second variation (V2) involves computing first precision, recall and F1 per lemma and then taking the average with respect to the total number of lemmas. Computing precision, recall and F1 per lemma helps in determining which lemmas our system is doing good at predicting the corresponding EWN synsets and which lemmas present a challenging situation to our system. For threshold tuning in each method, we

pick the threshold that achieves the best F1 on the training split. The threshold is varied from 0 to 1 with increments of 0.01. Based on V1, an overall F1 is computed as shown in equation 4.3 while based on V2, an average F1 is computed as in equation 4.6. True positives (TP) correspond to links that are present in the gold dataset and that were predicted as present by our model. False positives (FP) correspond to links that are absent from the gold dataset but were predicted as present by our method. False negatives (FN) correspond to the links that are present in the gold dataset but were predicted as absent by our method. True positives per lemma (TP_L) correspond to links with the Arabic lemma “L” that are present in the gold dataset and that were predicted as present by our model. False positive per lemma “L” (FP_L) correspond to links with the Arabic lemma “L” that are absent from the gold dataset but were predicted as present by our method. False negatives per lemma (FN_L) correspond to the links that are present in the gold dataset but were predicted as absent by our method. The threshold that leads to the highest Avg_F1 is utilized on the test data to assess the performance.

As NLP tools, we use the WordNet lemmatizer² available in the NLTK package in Python to perform lemmatization for English terms. The lemmatizer takes as input the English term and the POS tag. For stemming, we use Porter Stemmer³ also available in the NLTK package. English stop words are removed based on a list also provided by the NTLK package.

$$Precision = \frac{TP}{TP + FP} \quad (4.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4.3)$$

$$Avg_Precision = \frac{\sum Precision_L}{Total\#of\ Lemmas} = \frac{\sum \frac{TP_L}{TP_L + FP_L}}{Total\#of\ Lemmas} \quad (4.4)$$

$$Avg_Recall = \frac{\sum Recall_L}{Total\#of\ Lemmas} = \frac{\sum \frac{TP_L}{TP_L + FN_L}}{Total\#of\ Lemmas} \quad (4.5)$$

$$Avg_F1 = \frac{\sum F1_L}{Total\#of\ Lemmas} = \frac{\sum \frac{2 \times Precision_L \times Recall_L}{Precision_L + Recall_L}}{Total\#of\ Lemmas} \quad (4.6)$$

4.3.2 Direct Matching

The first set of approaches include matching SAMA English translation terms to EWN synset terms. A similarity score is computed for each pair of SAMA

²https://www.nltk.org/_modules/nltk/stem/wordnet.html

³https://www.nltk.org/_modules/nltk/stem/porter.html

lemma terms and EWN synset terms. Jaccard similarity was used in assessing the potential match since it helps measuring the overlap between two finite sets. Let A denote the Arabic lemma and G denote the set of its corresponding English translation terms. Let E denote an EWN synset and S denote its corresponding synset terms. The Jaccard similarity is computed as follows in equation 4.7.

$$\text{Direct_Matching_Similarity} = \frac{|G \cap S|}{|G \cup S|} \quad (4.7)$$

For example, the SAMA verb lemma, `xAlaf_v` (خالف), has three English translation terms: ‘contradict’; ‘conflict_with’; ‘go_against’. Given the EWN synset `02378851v` with synset terms ‘go_against#3’ ‘buck#2’ and gloss ‘resist’; ‘buck the trend’, the SAMA lemma/EWN synset pair represented by `xAlaf_v/02378851v` will have a Jaccard similarity of 25.00%.

We compute 4.7 after applying different NLP techniques on SAMA gloss terms and EWN synset terms. While EWN synset terms are lemmatized, SAMA gloss terms are not. Moreover, they include different special characters such as quotation marks, plus sign, and parentheses. Hence, we experiment with different settings on both sides.

We define the baseline to be computing Jaccard similarity using raw SAMA English translation terms and raw EWN synset (already lemmatized) terms. Jaccard similarity is computed between gold lemmas and all EWN synsets corresponding to the lemma’s POS tag. The remaining experimental settings are as follows:

0. Baseline: raw SAMA English translations vs. raw EWN synset terms.
1. Lemmatized SAMA gloss terms vs. raw EWN synset terms.
2. Applying Multiword handling (multiword splitting and stop words removal on lemmatized SAMA and EWN).
3. Stemmed SAMA English translation terms vs. stemmed EWN synset terms.
4. Multiword handling (multiword splitting and stop words removal on lemmatized SAMA and EWN) and then stemming both sides.

4.3.3 Using EWN Gloss and Extended Gloss

In this approach, we aim at increasing the recall of the previous approach by using a richer set of terms in EWN. We propose to use the gloss and extended gloss available with every synset in EWN. For example, the SAMA lemma `Hajom#n` (حجم) and its gloss terms: ‘size’; ‘volume’, is linked to the following EWN synset ‘05123416n’ with the synset term ‘extent#2’ and the gloss: ‘the distance or area or

volume over which something extends’; and the extended gloss ‘the vast extent of the desert’; ‘an orchard of considerable extent’. By just exploring SAMA English translation terms and EWN synset terms, we obtain a Jaccard similarity of zero. However, if we consider the EWN gloss and extended gloss, we observe that we can match SAMA English translation terms to EWN extended gloss terms. The approach incorporating the extended gloss is as follows:

1. Tokenize and lemmatize the gloss and the extended gloss
2. Remove stop words from the tokenized and lemmatized gloss and extended gloss
3. Compute Jaccard similarity between WordNet lemmatized SAMA gloss terms and the processed EWN gloss and extended gloss terms.

We also compute Jaccard similarity after applying the following processing variations:

1. Multiword handling (multiword splitting and stop words removal on lemmatized SAMA)
2. Stemming applied on both sides, SAMA and EWN
3. Multiword handling applied on SAMA side followed by stemming applied on both sides SAMA and EWN.

4.3.4 Using Machine Translation (MT) Tables

In order to address the challenge of limited SAMA gloss terms, we decided to benefit from existing aligned translated corpora [441] and their corresponding machine translation tables to expand the coverage of SAMA gloss terms. We decided to use MT tables instead of dictionaries since, 1) online dictionaries are not provided freely in a structured way that can be programmatically parsed, 2) although our main goal is to extend Arabic WordNet, we also want to develop a general approach that can also be applied successfully for other low resource languages where it is more common to find parallel corpora than dictionaries. The MT tables consist of two different tables. The first one includes three columns where the first column corresponds to Arabic lemmas, the second column is English terms that represent the translation of the Arabic lemmas based on alignment and the third column is a conditional probability score $p(Eng-Ar)$ that represents for each row the probability of having the English term given the Arabic lemma. The second table also includes three columns but the first column represents English terms, the second column represents the corresponding Arabic translation and the third column is also a conditional probability $p(Ar-Eng)$ that represents for each row in the table the probability

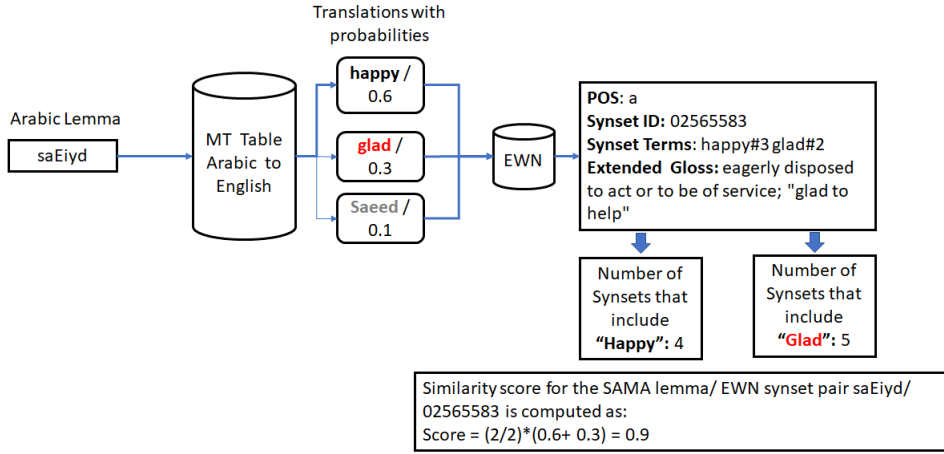


Figure 4.2: Walking example for computing similarity score between SAMA lemmas and EWN synsets based on MT tables.

of having the English word given the Arabic lemma. We use the Arabic to English table as a mean to predict the links between SAMA lemmas and EWN synsets.

The overall method consists of utilizing the MT tables to expand SAMA gloss terms and then applying a similarity measure against EWN synset terms. The method was only applied to lemmas that are in common between MT tables and SAMA. Around 7K SAMA lemmas do not appear in MT since these lemmas have a low frequency usage in MSA. Each Arabic lemma is represented by a set of English translations and a conditional probability score $P(Ar-Eng_i)$ is assigned for each Arabic lemma-English translation pair such that $\sum_i P(Ar|Eng_i) = 1$. In order to incorporate the probability information in the linking process, we come up with the following similarity measure (equation 4.8) for a given SAMA lemma-EWN synset pair where T represents the set of all possible English translations t_i of an Arabic lemma A and E the set of EWN synset terms e_i . We did not consider as denominator the cardinality of the union set of MT terms and EWN synset terms as is the case in equation 4.7 since MT terms tend to be significantly larger in terms of count compared to EWN synset terms. A simplified example is shown in Fig. 4.2.

$$MT_{similarity} = \frac{|T \cap E|}{|E|} \times \sum_{t_i \in \{T \cap E\}} P(A|t_i) \quad (4.8)$$

Similar to the methods described above, we also process the MT tables in different ways to try and improve the linking process. We compute the described similarity against EWN synsets with two different settings for MT English terms. We define the clean up process of MT English terms by: removing stop words,

punctuation and terms that include special characters such as (@, #, +, (,)) and then redistributing the probability scores to keep $\sum_i P(Ar|Eng_i) = 1$ satisfied. The different explored settings for performing link prediction using MT Tables are:

1. Using Raw MT English terms with EWN Synset terms
2. Lemmatizing MT English translations after the clean up process with EWN Synset terms.

In order to compare performance of using MT tables to when using SAMA with the same similarity function described above, we consider SAMA as a MT table and we assign equal probability for each gloss term for a given lemma. We then compute similarity scores with EWN synsets based on equation 4.8.

4.3.5 Using Word Embeddings

In order to address the challenge of different choices of words in SAMA versus EWN for the same meaning, we propose to use the state-of-the-art word embeddings [322] as a mean of semantic representation. Since synonym words would likely have similar vector representation since they can be used interchangeably, even if SAMA gloss terms and its corresponding EWN synset terms are different, their vector representation are expected to be similar. We use the cosine similarity measure to assess similarity between a SAMA lemma and an EWN synset. Since an Arabic lemma can have multiple English gloss terms and an EWN synset can consist of multiple terms we use the maximum, minimum and average cosine similarities. Assuming \vec{g}_i is the vector representation of a SAMA gloss term and \vec{s}_j is the vector representation of a synset term, the cosine similarity between a SAMA lemma with n gloss terms and an EWN synset with m synset terms can be computed as follows:

$$Average_Similarity = \frac{\sum_{i=1}^n \sum_{j=1}^m \frac{\vec{g}_i \cdot \vec{s}_j}{\|\vec{g}_i\| \times \|\vec{s}_j\|}}{m \times n} \quad (4.9)$$

$$Maximum_Similarity = \max\left(\frac{\vec{g}_i \cdot \vec{s}_j}{\|\vec{g}_i\| \times \|\vec{s}_j\|}\right) \quad (4.10)$$

$$Minimum_Similarity = \min\left(\frac{\vec{g}_i \cdot \vec{s}_j}{\|\vec{g}_i\| \times \|\vec{s}_j\|}\right) \quad (4.11)$$

We use the existing Google Word2Vec pretrained vector representations⁴. However, not all EWN terms and SAMA gloss terms have a vector representation in Google Word2Vec pretrained model due to different reasons. Terms in Word2Vec are not in their lemma form and Word2Vec does not

⁴<https://code.google.com/archive/p/word2vec/>

include vector representations for many multiword expressions. In order to address this issue, we split multiword expressions that do not have a vector representation into single terms and try to represent the expression as a sum of the vector representation of its single terms. Then the synset or lemma is represented by the centroid of the corresponding synset terms or SAMA English gloss terms. Cosine similarity is then computed between the centroid of SAMA lemma and all EWN synsets.

In order to make the computations efficient, mainly when using, Maximum, Minimum and Average, and to avoid computing similarity scores that are slightly greater than zero for links that will eventually be predicted as absent, we limit the computations to the EWN synsets that include terms from the Top 10 most similar English terms with respect to SAMA gloss terms. For example, for the SAMA lemma Hajar_n (حجر) with English gloss term: ‘stone’, we compute cosine similarity with all EWN synsets that include stone or at least one of the top 10 most similar words provided using Word2Vec package in Python: ‘stones’, ‘granite’, ‘marble’, ‘stone_slabs’, ‘bricks’, ‘marble_floorings’, ‘granite_stones’, ‘limestone’, ‘marble_slabs’ and ‘locally_quarried’. To further enhance the likelihood of matching the top 10 terms to EWN Synset terms, WordNet lemmatizer is applied on the top of the 10 terms if needed.

4.3.6 Fusion Model

In this subsection, we propose to fuse the different explored techniques into one model using three approaches. The first one consists of an aggregation of the normalized similarity measures of the different techniques. The result will be one similarity score for each potential link. A threshold will be determined as described in subsection 4.3.1. The second fusion approach consists of using a majority vote based on the decisions of the different techniques described above. Last but not least, the third fusion approach consists of using the computed similarity measures of the different explored techniques as features to one classification model. The similarity measures are first normalized and then used as attributes for a SVM classifier. We propose to utilize the concept of boosting in order to turn weak learners into a strong learner. Since the number of present links is much smaller than the number of absent links, we increase the cost of predicting actual present links as absent links. Given the sparsity of the training data when concatenating all the computed similarity features, we explore using PCA for sparsity reduction as well as Chi-square technique for feature reduction. We compare accuracy when using linear and non-linear SVM with Top 5, Top 8 (50% of the features), Top 10 and all of the similarity features, and with and without normalization.

4.4 Experimental Evaluation

We present in this section the results achieved for each approach when tested separately along with the corresponding error analysis. We then apply the fusion approaches as described in subsection 4.3.6 and evaluate the performance of the system on the test set. For each approach, a similarity threshold is learned to optimize F1 measure on the training set as described earlier in subsection 4.3.1. The threshold is then used to predict presence or absence of a link between a given SAMA lemma–EWN synset pair. Across all the methods explored, the same training and test sets are preserved to enable fair comparison among the different link prediction techniques. A description of the experimental setup used in the evaluation is first presented followed by the performance achieved for the different approaches.

4.4.1 Experimental Setup

The annotated data, called gold data set as discussed in subsection 4.1, consists of accurate links between SAMA lemmas and EWN synsets. The label “True” is assigned for all **present** links in the gold dataset while the label “False” is assigned to all **absent** links from the gold data set. As described in subsection 4.3.1, the data set is split into a training set (90%) and testing set (10%). The division is done randomly and is based on Arabic lemmas. For each POS tag, 90% of the lemmas were randomly selected along with their corresponding EWN links as training set. The remaining 10% were used as test set. The idea is to have a completely unseen test set, where there are no common lemmas between the training and the test sets. Detailed numbers about the count of lemmas, links, EWN synsets in the gold dataset can be found in Table 4.4.

4.4.2 Evaluation Approach

In order to evaluate the different techniques, performance results are reported based on two sets of equations. The first set consists of overall precision (4.1), overall recall (4.2), and overall F1 (4.3), and the second set consists of average precision per lemma (4.4), average recall per lemma (4.5), and average F1 per lemma (4.6). A comparison for the different variations within each major approach is shown in the upcoming subsections. Moreover, an error analysis is conducted by examining a sample of links that were misclassified. For each technique, we report the following:

1. The overall total number of links that were predicted as “True”, i.e., the number of True Positives and False Positives

2. The overall total number of False Negatives, i.e., the links that the technique labeled as absent while they are present in the gold data set
3. Within the False Negatives, deeper insights are considered by examining:
 - (a) The number of lemmas that did not link to any EWN synset, i.e., the similarity measure between the given lemma and any EWN synset is 0.
 - (b) The number of lemmas that linked to many EWN synsets, i.e. the similarity scores of the links are greater than 0, but none of those links are present in the gold dataset.

Those examples are inspected carefully in order to find the reasons behind the failure of each of the different techniques. The reasons are categorized and percentages of each category are reported.

4.4.3 Evaluation of Link Prediction Techniques Based on Direct Matching

As discussed in subsection 4.3.2, different processing variations are applied before computing the Jaccard similarity shown in equation 4.7. In Table 4.5, the results achieved on the test set are presented by computing equations 4.1, 4.2, and 4.3, where the threshold (Thr) is tuned to achieve the best overall F1 (equation 4.3) on the training set.

Tec	Matching Space	Noun				Verb				Adjective			
		Thr	Pre	Rec	F1	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1
(0)	Baseline	15	16.5	40.8	23.5	0	11.2	42.7	17.7	43	22.7	29.9	25.8
(1)	Lemmatization	14	15.7	50.0	23.9	0	11.2	42.7	17.7	43	22.5	29.9	25.6
(2)	Multiword Handling	34	12.9	18.4	15.1	34	8.2	11.4	9.5	43	21.7	29.9	25.2
(3)	Stemming	20	13.6	42.3	20.6	0	11.1	42.9	17.7	43	19.7	34.3	25.0
(4)	(2) followed by (3)	34	10.6	20.9	14.0	34	8.2	11.1	9.4	43	18.7	34.3	24.2

Table 4.5: Overall performance (%) of the different explored variations for Direct Matching per POS tag on the test set. Tec = Technique, Pre = Precision, Rec = Recall and Thr= Threshold.

In Table 4.6, average precision, recall and F1 per lemma are computed for the test set based on equations 4.4, 4.5 and 4.6. The threshold (Thr) is tuned to achieve the best Average_F1 shown in equation 4.6 on the training set. The threshold tuning is performed for each POS Tag separately.

Tables 4.7, 4.8 and 4.9 present additional insights about the impact of each variation of the direct matching approach on precision and recall for each of the three POS tags. Best F1 is achieved when using raw SAMA gloss terms for verbs and adjectives. For nouns, lemmatizing English SAMA gloss terms helped in slightly improving the performance compared to the baseline. Specifically,

Tec	Matching Space	Noun				Verb				Adjective			
		Thr	Pre	Rec	F1	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1
(0)	Baseline	5	21.5	50.5	26.4	0	13.8	41.0	18.0	29	17.9	37.4	22.7
(1)	Lemmatization	5	21.3	51.2	26.3	0	13.7	40.9	18.0	29	17.9	37.4	22.7
(2)	Multiword Handling	25	12.8	33.2	14.9	0	11.6	49.4	16.2	29	17.9	39.2	22.8
(3)	Stemming	5	17.7	54.5	23.4	0	13.7	41.1	17.9	29	18.7	42.7	23.8
(4)	(2) followed by (3)	25	10.8	35.4	13.3	0	11.6	50.0	16.2	29	17.9	42.7	23.0

Table 4.6: Average performance (%) per lemma of the different explored variations for Direct Matching per POS tag on the test set. Tec = Technique, Pre = Average Precision per lemma, Rec = Average Recall per lemma, F1 = Average F1 per lemma and Thr= Threshold.

lemmatization helped in improving the recall without causing a significant decrease in the precision. The analysis of tables 4.7, 4.8, and 4.9, show that multiword handling and stemming helped in reducing the number of lemmas with zero matches but at the same time significantly increased the number of lemmas that matched to synsets outside the gold dataset, thus, an increase in the count of false positives leading to poor performance. Hence, one can conclude that multiword handling and stemming should be used as a back off technique and not as a standalone method to ensure enhancement of recall without a significant drop in precision.

Technique	Matching Space	Links Count			Lemma Count (% of 445)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(0)	Baseline	373	1886	542	21 (4.7%)	148 (33.3%)
(1)	Lemmatization	457	2446	458	19 (4.3%)	121 (27.2%)
(2)	Multiword Handling	168	1136	747	3 (0.7%)	298 (67.0%)
(3)	Stemming	387	2459	528	15 (3.4%)	152 (34.2%)
(4)	(2) followed by (3)	191	1617	724	1 (0.2%)	283 (63.6%)

Table 4.7: Assessment of predictions on the test set for the Noun POS tag for the different variations of link prediction when using direct matching between SAMA gloss terms against EWN synset terms.

Technique	Matching Space	Links Count			Lemma Count (% of 214)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(0)	Baseline	224	1777	301	28 (13.1%)	41 (19.2%)
(1)	Lemmatization	224	1785	301	28 (13.1%)	42 (19.6%)
(2)	Multiword Handling	60	673	465	1 (0.5%)	149 (69.6%)
(3)	Stemming	225	1800	300	28 (13.1%)	42 (19.6%)
(4)	(2) followed by (3)	58	654	467	0	151 (70.6%)

Table 4.8: Assessment of predictions on the test set for the Verb POS tag for the different variations of link prediction when using direct matching between SAMA gloss Terms against EWN synset terms.

Technique	Matching Space	Links Count			Lemma Count (% of 57)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(0)	Baseline	20	68	47	4 (7.0%)	36 (63.2%)
(1)	Lemmatization	20	69	47	4 (7.0%)	36 (63.2%)
(2)	Multiword Handling	20	72	47	3 (5.3%)	37 (64.9%)
(3)	Stemming	23	94	44	1 (1.7%)	36 (63.2%)
(4)	(2) followed by (3)	23	100	44	1 (1.8%)	36 (63.2%)

Table 4.9: Assessment of predictions on the test set for the Adjective POS tag for the different variations of link prediction when using direct matching between SAMA gloss terms against EWN synset terms.

In order to understand the reasons behind the relatively low F1 achieved in the different variations, we conduct an error analysis on a sample of 300 SAMA lemma/EWN synset pairs from the test set. The 300 pairs were equally sampled from the three different POS tags. We analyzed samples generated when using raw SAMA English gloss terms since it performed best compared to other direct matching variations. 172 pairs are present in the gold dataset but were predicted as absent and 128 pairs are not in the gold dataset, i.e., their gold label is “False” but were predicted as present. The selection was designed to select boundary cases and included:

1. All gold links (26 for nouns, 53 for verbs and 4 for adjectives, so 27.67% of the 300 pairs) for lemmas that had a similarity score of 0 with all EWN synsets, i.e., the SAMA raw gloss terms of those lemmas did not match to any EWN synset term. The purpose of this selection is to try to identify reasons that explain why direct matching using raw gloss terms failed in predicting gold links as present. In terms of precision and recall, the analysis of such examples helps in recognizing recall issues.
2. Many lemmas only linked to EWN synsets that do not appear in the gold dataset. We randomly picked a set of these lemmas and looked at their corresponding predicted links, and at their corresponding gold links. In total, we had 74 pairs for nouns, 47 pairs for verbs and 96 pairs for adjectives, so 72.33% of the 300 pairs. The purpose of this selection is to try to identify the reasons for predicting links as present while the links are absent from the gold set. In terms of precision and recall, the analysis of such examples helps in understanding recall and precision issues but mainly in potentially improving the precision of our link prediction system.

In order to perform quantitative analysis, we split the samples analyzed into two sets: the first set consists of links that exist in the gold dataset but were predicted to have a “False” label (False Negatives) and the second set consists of links that were predicted as “True” but are absent from the gold dataset (False Positives). The results are summarized in Fig. 4.3.



Figure 4.3: Distribution of error reasons when using direct matching for false negatives and false positives by POS tag. FN1: Different Choices of English Words in SAMA versus EWN to Represent the Same Meaning. FN2: Limited SAMA English Gloss Terms. FN3: Different Inflections Used to Represent the Same English Lemma. FN4: Not Satisfying Minimum Threshold. FN5: Multiword Terms. FN6: Mismatch between AWN Lemma and SAMA Lemma. FP1: Matching to Homonyms between SAMA English Gloss Terms and EWN Synset Terms. FP2: Lack of Comprehensive Coverage of Links in the Gold Set. FP3: Mismatch between AWN Lemma and SAMA Lemma.

4.4.3.1 Analysis of False Negatives

The reasons behind predicting gold links as absent are described in what follows. We labeled the reasons from FN1 to FN6 based on the frequency of each reason.

1. **FN1: Different Choices of English Words in SAMA versus EWN to Represent the Same Meaning:** This reason happens around 39% in total across all POS tags. As described in subsection 4.2, SAMA English gloss terms and EWN synset terms are semantically linked, i.e., have the

same meaning, but the terms used in one lexicon are different from the terms in the other. Hence, direct matching technique fails at identifying them as “True” links.

- (a) As a noun example: the Arabic lemma *sumuw~* (سمو) with SAMA English gloss terms, ‘his/her highness’, is linked in the gold set to two EWN synsets. The first synset, 04814238, includes as synset terms: ‘magnificence’, ‘brilliance’, ‘grandeur’, ‘grandness’, ‘splendour’, ‘splendor’, and has as gloss and extended gloss: ‘the quality of being magnificent or splendid or grand’; ‘for magnificence and personal service there is the Queen’s hotel’; ‘his ‘Hamlet’ lacks the brilliance that one expects’; ‘it is the university that gives the scene its stately splendor’; ‘an imaginative mix of old-fashioned grandeur and colorful art’; ‘advertisers capitalize on the grandness and elegance it brings to their products’. The second synset, 14431902, includes as synset terms: ‘dignity’, and has as gloss and extended gloss: ‘high office or rank or station’; ‘he respected the dignity of the emissaries’.
- (b) As a verb example: the Arabic lemma *Ealiq* (علق) has as English gloss terms: ‘be attached’. *Ealiq* is linked in the gold set to two EWN synsets. The first synset, 02734952, includes as synset terms: ‘hang’, and has as gloss and extended gloss: ‘be menacing, burdensome, or oppressive’; ‘This worry hangs on my mind’; ‘The cloud of suspicion hangs over her’. The second synset, 0120749, includes as synset terms: ‘immobilize’, ‘immobilise’, ‘trap’, ‘pin’, and has as gloss and extended gloss: ‘to hold fast or prevent from moving’; ‘The child was pinned under the fallen tree’. It is also important to note that some EWN synsets include terms that are not in English with United States spelling but in other spelling or Latin languages such as French as is the case of the terms ‘immobilise’ and ‘qui vive’. The existence of such terms has an impact on decreasing the value of Jaccard similarity.
- (c) The adjective SAMA lemma **awoqiy~* (ذوقِي) showcases how the English word choice of SAMA is different than EWN. **awoqiy~* has as SAMA English gloss terms ‘sense’, ‘taste’ and it should be linked to the EWN synset 02868916, with synset terms: ‘gustatorial’, ‘gustatory’, ‘gustative’, with gloss and extended gloss: ‘of or relating to gustation’. In addition to word choice, we notice here that the POS of the SAMA gloss terms do not always match with the POS of the corresponding Arabic lemma. For this case, nouns (‘sense’ and ‘taste’) are used to describe an adjective in SAMA English gloss terms.

2. **FN2: Limited SAMA English Gloss Terms:** FN2 occurs around 36% in total for all POS tags. In other words, some semantic meanings of a given Arabic SAMA lemma are missing from English SAMA gloss terms. For example:
- (a) The noun lemma Hasab (حسب) with English gloss terms: ‘according to’, ‘according to + [def.acc.] + what’, ‘in accordance with’, only includes one possible meaning of the Arabic lemma. Hasab is linked in the gold dataset to the EWN synset, 14431902, with synset terms: ‘laurels’, ‘honor’, ‘honour’, and with gloss and extended gloss: ‘the state of being honored’.
 - (b) Similarly for verbs, the SAMA lemma >aqoEad (أقعد) with gloss terms: ‘be made stay’, ‘be sat down’, ‘make sit’, ‘make stay’, should be linked to the EWN synset, 00091968, with synset terms: ‘handicap’, ‘invalid’, ‘incapacitate’, ‘disable’, and with gloss and extended gloss: ‘injure permanently’; ‘He was disabled in a car accident’.
 - (c) As an adjective example, the lemma nawoEiy~ (نوعي) has SAMA English definitions: ‘characteristic’, ‘specific’, ‘type’. The corresponding EWN synset, 01914521, has as synset terms: ‘qualitative’, and as gloss and extended gloss: ‘involving distinctions based on qualities’; ‘qualitative change’; ‘qualitative data’; ‘qualitative analysis determines the chemical constituents of a substance or mixture’.
3. **FN3: Different Inflections Used to Represent the Same English Lemma:** Words used in SAMA English gloss terms and EWN synset terms do not have the same inflection. FN3 occurs 11% of the times. For this case, stemming or lemmatization can address this issue. However, it may not work in all cases. For example, the noun SAMA lemma >urovuwdusiy~ (أرثودكسي) with English gloss term: orthodox should be linked to the EWN synset 04801313, with synset terms: ‘orthodoxy’, and extended gloss: ‘the quality of being orthodox (especially in religion)’. Using Porter stemmer, the stem of ‘orthodoxy’ is ‘orthodoxi’ and not ‘orthodox’, thus using direct matching with stemming would not help in predicting the link as present. But, by matching SAMA gloss terms to EWN extended gloss, the link may be predicted as present.
4. **FN4: Not Satisfying Minimum Threshold:** In some cases, although the Jaccard similarity of some links were greater than 0, the links were predicted as absent since the corresponding similarity scores were below the optimum threshold. FN4 occurs 8% in total out of the 172 samples. For

example, the adjective SAMA lemma nADij (ناضج) with SAMA gloss terms: ‘mature’, ‘ripe’, ‘well-cooked’, has a Jaccard similarity of 33.33% with its corresponding EWN synset, 01488245, that has as synset terms: ‘mature’, and as extended gloss: ‘having reached full natural growth or development’; ‘a mature cell’. 33.33% is less than 43%, the optimal threshold that achieved the best F1 on the training set of adjectives.

5. **FN5: Multiword Terms:** Multiword terms occur in both English SAMA gloss terms and EWN synset terms. When multiword terms are split, the resulting terms can match to single terms. FN5 occurs 4% in total. For example, the noun SAMA lemma junayoh (جنيه) has as SAMA English gloss terms: ‘pound (currency)’, ‘pounds (currency)’. The corresponding EWN synset 13694017 has as synset terms ‘egyptian pound’, ‘pound’ and as extended gloss: ‘the basic unit of money in Egypt’; ‘equal to 100 piasters’. When applying multiword splitting, the link will be predicted as present.
6. **FN6: Mismatch between AWN Lemma and SAMA Lemma:** In some cases, due to the automated part for mapping AWN lemmas to SAMA lemmas, where we automatically assigned the closest match generated by ALMOR to a given AWN lemma, some links in the gold dataset are not accurate. FN 6 occurs 2% in total.
 - (a) For instance, the noun AWN lemma qaA}ilap (قائلة) was automatically mapped to qA}il (قائل) in SAMA when using ALMOR since qaA}ilap is not present in SAMA. However, those two lemmas are different in meanings and qaA}ilap is not the feminine form of qaA}il. The SAMA lemma qaA}il has as English definitions: ‘person who says’, ‘sayer’. The wrong corresponding EWN synset to qaA}ilap, 15165490, has as synset terms: ‘high noon’, ‘twelve noon’, ‘noon’, ‘noontide’, ‘midday’, ‘noonday’, and as extended gloss: ‘the middle of the day’.
 - (b) Another example is the AWN lemma jaraAbiy (جراي) that was mistakenly mapped to the SAMA lemma jurAb (جراب) since jaraAbiy is not present in SAMA and the closest match provided by ALMOR was jurAb.

Given the low percentage of such cases, we will ignore this error.

4.4.3.2 Analysis of False Positives

There are mainly three reasons for classifying a link as present between a SAMA lemma and an EWN synset while the link is absent from the gold data. We labeled the reasons as FP1 to FP3 for easier reference in the figures.

1. **FP1: Matching to Homonyms between SAMA English Gloss Terms and EWN Synset Terms:**

FP1 happens 54% of the times. In some cases, a SAMA lemma may have a Jaccard similarity with a given EWN above the optimal threshold. However, the meaning reflected by the Arabic lemma does not correspond to the one of EWN synset. The link is absent from the gold dataset, but it is predicted as present using direct matching approach.

(a) As for example, the noun SAMA lemma tano\$iyT (تنشيط) has as English gloss terms: encouragement, stimulation. A Jaccard similarity of 50% is obtained between tano\$iyT and the EWN synset, 06691442, with synset terms: ‘encouragement’, and extended gloss: ‘the expression of approval and support’. However, the meaning is different between the SAMA lemma and the EWN synset. The correct Arabic lemma is ta\$jjiE (تشجيع).

(b) Another example is the verb SAMA lemma >aqoSaY (أقصى) with English SAMA gloss terms: ‘be removed’, ‘remove’. A Jaccard similarity of 50% is obtained between >aqoSaY and the EWN synset, 02224055, with synset terms: ‘remove’, ‘get rid of’, and with extended gloss: ‘dispose of’; ‘get rid of these old shoes!’; ‘the company got rid of all the dead wood’. Although the Jaccard similarity is above the optimal threshold, the meaning of the SAMA lemma does not correspond to the one expressed by the EWN synset. The correct Arabic lemma is taxal~aS (تخلص).

(c) The adjective lemma musotaEid~ (مستعد) has as English gloss terms: ‘prepared’, ‘ready’. A Jaccard similarity of 50% is obtained between musotaEid~ and the EWN synset, 00185759, with synset terms: ‘ready’, and with extended gloss: ‘(of especially money) immediately available’; ‘he seems to have ample ready money’; ‘a ready source of cash’. Again, the SAMA lemma is not the accurate meaning of the EWN synset, instead the adjective jAhiz (جاهز) is.

2. **FP2: Lack of Comprehensive Coverage of Links in the Gold Set:**

By examining some of the links that are false positives, we found out that those links are actually correct in terms of meaning. We refer to such links as missing true positives. Since AWN is not complete, having such links is expected. FP2 happens 45% in total out of the 128 analyzed samples.

(a) For example, a link was predicted as present between the SAMA lemma suEor (سعر) with English gloss terms: ‘madness’ and the EWN synset, 07516997, that has as synset terms: ‘fury’, ‘madness’,

‘rage’, and as extended gloss: ‘a feeling of intense anger’; ‘hell hath no fury like a woman scorned’; ‘his face turned red with rage’. The link is true, but it is not present in the gold dataset. In the gold dataset, suEor was linked to the EWN synset, 13726296, with synset terms: ‘calorie’, ‘small calorie’, ‘gram calorie’, and with extended gloss: ‘unit of heat defined as the quantity of heat required to raise the temperature of 1 gram of water by 1 degree centigrade at atmospheric pressure’.

- (b) Another example is the adjective SAMA lemma mariyr (مريير) with English gloss terms: ‘bitter’, ‘stubborn’. A link was predicted as present between mariyr and the EWN synset, 01364993, with synset term: ‘bitter’, and with extended gloss: ‘expressive of severe grief or regret’; ‘shed bitter tears’. The link is true, but it is not present in the gold dataset. In the gold dataset, the adjective mariyr is linked to the EWN synset, 00648614, with synset terms: ‘vituperative’, ‘scathing’, and with extended gloss: ‘marked by harshly abusive criticism’; ‘his scathing remarks about silly lady novelists’; ‘her vituperative railing’. Due to word choice, the latter gold link was predicted as absent.
- (c) Another example is the verb SAMA lemma >avonaY (أثنى) with English SAMA gloss terms: ‘be commended’, ‘be praised’, ‘commend’, ‘praise’. A link was predicted as present between >avonaY and the EWN synset, 01689169, with synset terms: ‘commend’ and extended gloss: ‘present as worthy of regard, kindness, or confidence’; ‘His paintings commend him to the artistic world’. Although the link is not present in the gold dataset, it is true in terms of meaning. In the gold dataset the verb ʔavonaY was linked to the EWN synset, 00860620, with synset terms: ‘glorify’, ‘laud’, ‘extol’, ‘proclaim’, ‘exalt’, and extended gloss: ‘praise, glorify, or honor’; ‘extol the virtues of one’s children’; ‘glorify one’s spouse’s cooking’. Due to word choice difference, the gold link was predicted as absent although ‘commend’, ‘laud’ and ‘extol’ are synonyms. Another example is the verb SAMA lemma >aqoSaY (أقصى) with English SAMA gloss terms: ‘be removed’, ‘remove’. A Jaccard similarity of 50% is obtained between >aqoSaY and the EWN synset, 02404224, with synset terms: ‘remove’, and with extended gloss: ‘remove from a position or an office’.

3. **FP3: Mismatch between AWN Lemma and SAMA Lemma:** Given the wrong mapping that occurred in very few cases between AWN and SAMA as mentioned in FN6, some of those lemmas linked to synsets in

EWN that are not in the gold dataset. FP3 happens only 1% of the times. While the links are obviously not in the gold dataset, some of the links are actually true in terms of meaning between Arabic and English. As an example, the AWN noun lemma kaniyosap (كنيسة) was wrongly mapped to kaniys (كنيس) in SAMA. kaniys has as English gloss terms: ‘synagogue’, ‘synagogues’, ‘temple’, ‘temples’. As a result, kaniys was linked to the EWN synset, 04374735, with EWN synset terms: ‘tabernacle’, ‘temple’, ‘synagogue’ and with extended gloss: ‘(Judaism) the place of worship for a Jewish congregation’. While the link is not in the gold dataset, the meaning of the SAMA lemma is accurately represented by the EWN synset. The corresponding EWN synset for the lemma kaniyosap is 03028079 with synset terms ‘church’, ‘church building’ and with extended gloss: ‘a place for public (especially Christian) worship’, ‘the church was empty’.

4.4.4 Evaluation of Link Prediction Techniques Based on EWN Gloss & Extended Gloss

As discussed in subsection 4.3.3, different processing variations are applied on SAMA gloss terms and EWN gloss and extended gloss before computing Jaccard similarity between SAMA gloss terms and EWN gloss and extended gloss terms. In Table 4.10, the results achieved on the test set are presented by computing overall precision, recall and F1 using equations 4.1, 4.2, and 4.3 respectively, where the threshold (*Thr*) is tuned to achieve the best overall F1 (equation 4.3) on the training set. In Table 4.11, average precision, recall and F1 per lemma are computed for the test set based on equations 4.4, 4.5 and 4.6 respectively. The threshold (*Thr*) is tuned to achieve the best Average_F1 shown in equation 4.6 on the training set. The threshold tuning is performed for each POS Tag separately. Tables 4.7, 4.8 and 4.9 present additional insights for each of the three POS tags, about the impact of each set of processing steps applied on EWN gloss and extended gloss, on precision and recall. Based on the results, one can conclude that using EWN gloss and extended gloss matched against SAMA English gloss terms as a standalone technique introduces a lot of faulty links resulting in a very low performance. However, the technique can be helpful in improving the recall of direct matching approaches by addressing some of the issues highlighted in the error analysis of direct matching approaches in section 4.4.3.

4.4.5 Evaluation of Link Prediction Techniques Using MT Tables

As discussed in subsection 4.3.4, an Arabic to English MT table is utilized for link prediction. In addition to using raw MT tables, lemmatization is also applied on MT English terms before computing the similarity measure

Tec	Matching Space	Noun				Verb				Adjective			
		Thr	Pre	Rec	F1	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1
(1)	Lemmatization	22	1.3	3.8	2.0	0	2.4	42.5	4.5	12	1.0	14.9	1.9
(2)	Multiword Handling	22	1.1	4.0	1.8	22	2.3	5.1	3.2	13	1.0	10.5	1.8
(3)	Stemming	22	1.4	5.4	2.2	0	2.1	43.6	4.1	13	1.0	19.4	1.9
(4)	(2) followed by (3)	25	1.4	2.2	1.7	20	2.3	6.9	3.4	13	1.0	19.4	1.9

Table 4.10: Overall performance (%) of the different explored variations for link prediction using EWN Gloss and Extended Gloss. Tec = Technique, Pre = Precision, Rec = Recall and Thr= Threshold.

Tec	Matching Space	Noun				Verb				Adjective			
		Average				Average				Average			
		Thr	Pre	Rec	F1	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1
(1)	Lemmatization	3	1.0	25.9	1.8	3	3.9	38.3	6.5	5	4.8	38.3	6.1
(2)	Multiword Handling	3	1.0	29.2	1.8	4	3.5	47.5	5.8	5	4.8	40.1	6.1
(3)	Stemming	15	1.6	10.4	2.1	0	3.4	39.4	5.8	5	4.5	45.3	5.5
(4)	(2) followed by (3)	15	1.6	11.3	2.2	3	3.0	49.9	5.2	5	4.6	47.1	5.5

Table 4.11: Average performance (%) per lemma of the different explored variations for link prediction using EWN gloss and extended gloss. Tec = Technique, Pre = Average Precision per lemma, Rec = Average Recall per lemma, F1 = Average F1 and Thr= Threshold.

Technique	Matching Space	Links Count			Lemma Count (% of 445)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	Lemmatization	35	2639	880	29 (6.5%)	365 (82.0%)
(2)	Multiword Handling	37	3233	878	9 (2.0%)	379 (85.2%)
(3)	Stemming	49	3580	866	24 (5.4%)	353 (79.3%)
(4)	(2) followed by (3)	20	1403	895	5 (1.1%)	410 (92.1%)

Table 4.12: Assessment of predictions on the test set for the Noun POS tag for the different variations of link prediction of using SAMA gloss terms against EWN gloss & extended gloss.

Technique	Matching Space	Links Count			Lemma Count (% of 214)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	Lemmatization	223	9275	302	33 (15.4%)	39 (18.2%)
(2)	Multiword Handling	27	1159	498	4 (1.9%)	173 (80.8%)
(3)	Stemming	229	10454	296	33 (15.4%)	37 (17.3%)
(4)	(2) followed by (3)	36	1544	489	0	162 (75.7%)

Table 4.13: Assessment of predictions on the test set for the Verb POS tag for the different variations of link prediction of using SAMA gloss terms against EWN gloss & extended gloss.

presented in equation 4.8. The choice of the pre-processing was selected based on the best performing variation when using direct matching. To compare the MT table to SAMA, we have also treated SAMA as an MT and assigned probability scores to the English SAMA gloss terms as described in

Technique	Matching Space	Links Count			Lemma Count (% of 57)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	Lemmatization	10	994	57	4 (7.0%)	41 (71.9%)
(2)	Multiword Handling	7	714	60	4 (7.0%)	43 (75.4%)
(3)	Stemming	13	1302	54	2 (3.5%)	40 (70.2%)
(4)	(2) followed by (3)	13	1342	54	2 (3.5%)	39 (68.4%)

Table 4.14: Assessment of predictions on the test set for the Adjective POS tag for the different variations of link prediction of using SAMA gloss terms against EWN gloss & extended gloss.

subsection 4.3.4. In Table 4.15, the results achieved on the test set are presented by computing equations 4.1, 4.2, and 4.3, where the threshold (Thr) is tuned to achieve the best overall F1 (equation 4.3) on the training set.

Tec	Matching Space	Noun				Verb				Adjective			
		Thr	Pre	Rec	F1	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1
(1)	No modification	10	25.4	33.1	28.8	10	16.6	15.8	16.2	22	22.6	35.8	27.8
(2)	Lemmatization	14	25.2	32.5	28.4	18	15.4	24.4	18.9	22	22.6	35.8	27.8
(3)	SAMA as MT	7	15.7	50.0	23.9	1	11.2	42.7	17.7	7.0	13.8	41.8	20.7

Table 4.15: Overall performance (%) of the different explored variations for link prediction using MT tables. Tec = Technique, Pre = Precision, Rec = Recall and Thr= Threshold.

In Table 4.16, average precision, recall and F1 per lemma are computed for the test set based on equations 4.4, 4.5 and 4.6. The threshold (Thr) is tuned to achieve the best Average_F1 shown in equation 4.6 on the training set. The threshold tuning is performed for each POS Tag separately. Tables 4.17, 4.18 and 4.19 present additional insights about the impact of each variation of using MT tables on precision and recall for each of the three POS tags. For nouns, using raw MT tables performed best, while for verbs, lemmatization helped improving overall F1 by 2.6%. For adjectives, there was no change in the performance. Using MT resulted in better overall F1 compared to when using SAMA although some Arabic lemmas do not have an MT representation. For instance, 134 noun lemmas out of 4,452 lemmas (3.01%) in the gold set do not have a MT representation, for verbs 114 out of 2,153 lemmas (5.29%), and for adjectives 13 out of 576 lemmas (2.26%).

4.4.5.1 Comparison with Direct Matching

In terms of F1, using MT with lemmatization achieved better overall F1 compared to the best scores achieved when using direct matching with SAMA English gloss terms. We also looked at the number of lemmas with zero links and the number of lemmas that had links with synsets outside the gold set. We compared the counts of those lemmas when using raw MT versus when using baseline of direct

		Noun				Verb				Adjective			
		Average				Average				Average			
Tec	Matching Space	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1
(1)	No modification	3	18.1	49.9	23.4	3	11.3	29.6	14.2	10	18.2	46.8	24.2
(2)	Lemmatization	10	23.2	39.9	26.0	12	14.1	27.2	16.5	10	18.0	46.8	24.0
(3)	SAMA as MT	2	21.3	51.2	26.3	1	13.7	40.9	18.0	4	15.5	43.3	20.3

Table 4.16: Average performance (%) per lemma of the different explored variations for link prediction using MT tables. Tec = Technique, Pre = Average Precision per lemma, Rec = Average Recall per lemma, F1 = Average F1 per lemma and Thr= Threshold.

		Links Count			Lemma Count (% of 445)	
Technique	Matching Space	True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	No Modification	10	994	57	4 (7.0%)	41 (71.9%)
(2)	Lemmatization	7	714	60	4 (7.0%)	43 (75.4%)
(3)	SAMA as MT	13	1302	54	2 (3.5%)	40 (70.2%)

Table 4.17: Assessment of predictions on the test set for the Noun POS tag for the different variations of link prediction when using MT Tables against EWN synset terms.

		Links Count			Lemma Count (% of 214)	
Technique	Matching Space	True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	No Modification	83	417	442	19 (8.9%)	141 (65.9%)
(2)	Lemmatization	128	702	397	15 (7.0%)	112 (52.3%)
(3)	SAMA as MT	224	1779	301	28 (13.1%)	42 (19.6%)

Table 4.18: Assessment of predictions on the test set for the Verb POS tag for the different variations of link prediction when using MT Tables against EWN synset terms.

		Links Count			Lemma Count (% of 57)	
Technique	Matching Space	True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	No Modification	24	82	43	0	27 (47.4%)
(2)	Lemmatization	24	82	43	0	27 (47.4%)
(3)	SAMA as MT	28	175	39	4 (7.0%)	25 (43.9%)

Table 4.19: Assessment of predictions on the test set for the Adjective POS tag for the different variations of link prediction when using MT Tables against EWN synset terms.

matching. In order to check whether MT helped in addressing some of the issues of direct matching, we evaluated the number of common lemmas for each of the two cases. For the case of zero links, the number represents the limitations of both techniques even if fused together. Similarly for the case of linking only to synsets outside the gold dataset. By looking at Fig. 4.4, we notice that MT has a lower number of lemmas that got a similarity score of 0 although the count

for MT includes the lemmas that do not have a MT representation as well. The reduction in the number is expected since many Arabic lemmas have a large number of possible translation alignments. For instance, the Arabic noun lemma {inofijAr (إنفجار) has 476 possible alignments, the Arabic verb lemma {iEotabar (اعتبر) has 1160 possible alignments, and the Arabic adjective lemma baEiyd (بعيد) has 555 probable alignments. The same argument justifies the increase in the number of lemmas with links only to synsets outside the gold dataset. The number of common lemmas showcase the limitation of both techniques even if combined. The significant number of lemmas having links to outside the gold dataset has an impact on recall since we are missing the links that are in the gold dataset and on the precision since we are assuming the gold label of all the links outside the gold dataset to be “False”. As seen in the analysis of the sample of False positives in subsection 4.4.3.2, 45% consisted of missing true positives.

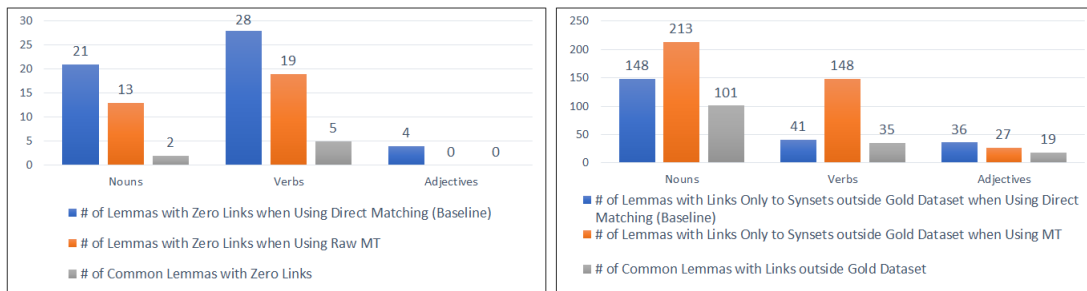


Figure 4.4: Comparison of count of lemmas with zero links and count of lemmas with links to EWN synsets outside the gold dataset in the test set when using raw MT and direct matching with raw SAMA English gloss terms.

We have also looked at the lemmatized variation for each of the two techniques, i.e. using direct matching and using MT tables. Fig. 4.5 shows the corresponding results. We notice that for nouns the number of lemmas that had a similarity score of zero with all EWN synsets or that only linked to synsets outside the gold dataset decreased in both techniques. This is the case for verbs as well but only when using MT. This indicates that MT representation for verbs is better than SAMA representation where there is a tendency in SAMA gloss terms to use a phrase to represent the meaning of the verb. For example the verb nak~ ad (نَكَّدَ) has as SAMA English gloss terms: ‘make life difficult’, while its corresponding EWN synset in the gold set has as synset terms: ‘chevy’, ‘molest’, ‘chivy’, ‘chivvy’, ‘plague’, ‘harass’, ‘provoke’, ‘hassle’, ‘beset’, ‘chevvy’, ‘harry’. For adjectives, there were no changes.

By specifically looking at the random sample consisting of 300 links analyzed for direct matching experiment, we observe enhancements when using MT. For nouns and when using direct matching, 24 gold links were predicted as absent

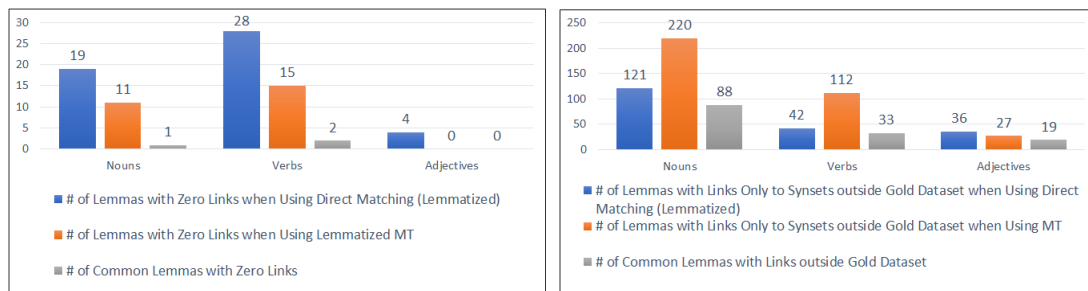


Figure 4.5: Comparison of count of lemmas with zero links and count of lemmas with links to EWN synsets outside the Gold dataset in the test set when using lemmatized MT and direct matching with lemmatized SAMA English gloss terms.

since the corresponding lemmas had a zero similarity score with all EWN synsets (case 1). 7 of those links (29.17%) were correctly labeled when using MT. 41 gold links when using direct matching were predicted as absent since the corresponding lemmas had similarity score above optimum threshold only with EWN synsets outside the gold dataset (case 2). Also 7 of those 41 links (17.07%) were correctly labeled when using MT. For verbs, 10 out of 53 (18.87%) golds links belonging to case 1 were accurately predicted when using MT, and 3 out of 22 gold links (13.64%) from case 2 were predicted accurately as present when using MT. For adjectives, none of the gold links (0%) falling under case 1 were fixed when using MT, and 4 out of 22 gold links (18.18%) belonging to case 2 were predicted as present when using MT.

4.4.6 Evaluation of Link Prediction Techniques Based on Word Embedding

As discussed in subsection 4.3.5, word embeddings are utilized to compute cosine similarity between SAMA English gloss terms and EWN Synset terms for link prediction. Average, maximum, minimum and centroid similarity measures are computed. In Table 4.20, the results achieved on the test set are presented by computing equations 4.1, 4.2, and 4.3 for the different variations of cosine similarity with word embeddings, where the threshold (Thr) is tuned to achieve the best overall F1 (equation 4.3) on the training set.

In Table 4.21, average precision, recall and F1 per lemma are computed for the test set based on equations 4.4, 4.5 and 4.6. The threshold (Thr) is tuned to achieve the best Average.F1 shown in equation 4.6 on the training set. The threshold tuning is performed for each POS Tag separately.

Tables 4.22, 4.23 and 4.24 present additional insights about the impact of using the average, maximum and minimum on the precision and recall for each of the three POS tags. In terms of overall F1, using maximum performed best for nouns

Tec	Matching Space	Noun				Verb				Adjective			
		Thr	Pre	Rec	F1	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1
(1)	Average	76	13.9	43.3	21.1	78	19.5	26.7	14.0	75	10.4	41.8	16.6
(2)	Maximum	98	15.3	52.2	23.7	89	11.0	42.9	17.6	97	9.4	44.8	15.6
(3)	Minimum	65	8.9	28.4	13.6	86	9.1	12.00	10.3	97	7.3	7.5	7.4
(4)	Centroid	80	15.6	20.1	17.6	71	8.7	16.2	11.3	83	20.6	22.4	21.4

Table 4.20: Overall performance (%) of the different explored variations for link prediction using Word Embeddings. Tec = Technique, Pre = Precision, Rec = Recall and Thr= Threshold.

Tech	Matching Space	Noun				Verb				Adjective			
		Average				Average				Average			
		Thr	Pre	Rec	F1	Thr	Pre	Rec	F1	Thr	Pre	Rec	F1
(1)	Average	82	14.2	28.3	26.2	83	9.2	18.0	10.4	84	5.6	10.5	6.7
(2)	Maximum	98	21.2	51.2	26.3	89	13.4	41.1	17.9	97	15.4	43.3	20.1
(3)	Minimum	98	9.1	15.8	9.9	89	7.3	11.4	7.2	97	3.5	7.0	4.3
(3)	Centroid	92	7.4	8.4	7.1	64	11.6	28.5	14.1	68	14.3	34.8	18.5

Table 4.21: Average performance (%) per lemma of the different explored variations for link prediction using Word Embeddings. Tech = Technique, Pre = Average Precision per lemma, Rec = Average Recall per lemma, F1 = Average F1 per lemma and Thr= Threshold.

and verbs, while using centroid performed best for adjectives. When looking at average F1 per lemma, using maximum performed best for all pos tags. The number of lemmas that did not link to any EWN synset decreased compared to baseline and reached 0 for verbs. This indicates that very few lemmas of the gold test set had English gloss terms that do not appear in Google Word2Vec vocab. It is important to note that for multiword expressions, we have tried first if the expression appears in Google Word2Vec vocab. If there were no representation for the expression, we split the multiword into single terms and we summed the vector representations of the terms and looked for the top 10 similar words given the resulting vector.

Technique	Matching Space	Links Count			Lemma Count (% of 445)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	Average	396	2449	519	6 (1.4%)	153 (34.4%)
(2)	Maximum	478	2643	437	2 (0.5%)	128 (28.8%)
(3)	Minimum	260	2662	655	6 (1.4%)	222 (49.9%)
(4)	Centroid	184	998	731	5 (1.1%)	291 (65.4%)

Table 4.22: Assessment of predictions on the test set for the Noun POS tag for the different variations of link prediction when using Word Embeddings between SAMA gloss terms and EWN synset terms.

We also compare the number of lemmas that did not match to any synset when using word embedding maximum and when using direct matching with lemmatization. Similarly, we compare the number of lemmas from the test set

Technique	Matching Space	Links Count			Lemma Count (% of 214)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	Average	140	1341	385	0	98 (45.8%)
(2)	Maximum	225	1813	300	0	69 (32.2%)
(3)	Minimum	63	632	462	0	157 (73.4%)
(4)	Centroid	85	897	440	0	132 (61.7%)

Table 4.23: Assessment of predictions on the test set for the Verb POS tag for the different variations of link prediction when using Word Embeddings between SAMA gloss terms and EWN synset terms.

Technique	Matching Space	Links Count			Lemma Count (% of 57)	
		True Positive	False Positive	False Negative	Zero Similarity	Links Not in Gold
(1)	Average	28	242	39	3 (5.3%)	27 (47.4%)
(2)	Maximum	30	288	37	3 (5.3%)	26 (45.6%)
(3)	Minimum	5	64	62	3 (5.3%)	49 (86.0%)
(4)	Centroid	15	58	52	1 (1.8%)	44 (77.2%)

Table 4.24: Assessment of predictions on the test set for the Adjective POS tag for the different variations of link prediction when using Word Embeddings between SAMA gloss terms and EWN synset terms.

that linked only to synsets outside the gold dataset when using word embedding maximum and when using direct matching. The results are shown in Fig. 4.6. As can be seen in Fig. 4.6, the number of lemmas with zero links decreased when using word embeddings resulting in an increase in recall for all POS tags compared to when using direct matching. On the other hand, the number of lemmas that linked only to synsets outside the gold dataset increased justifying the slight decrease in precision when using word embeddings compared to when using direct matching with lemmatization.

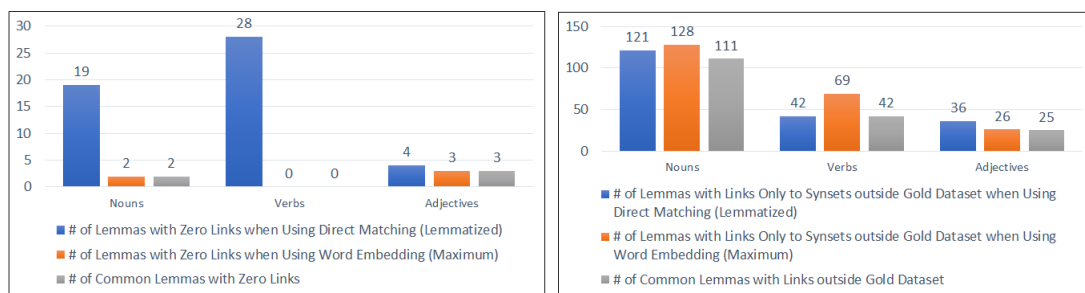


Figure 4.6: Comparison of count of lemmas with zero links and count of lemmas with links to EWN synsets outside the gold dataset in the Test set when using Word Embedding Maximum and direct matching with lemmatized SAMA English gloss terms.

By looking specifically at the random sample of 300 links from the test set,

using word embeddings maximum helped in predicting accurately some of the links with gold label “True”. For nouns and when using direct matching, 24 gold links were predicted as absent since the corresponding lemmas had a zero similarity score with all EWN synsets (case 1). None of those links (0%) were correctly labeled when using word embedding maximum. 41 gold links when using direct matching were predicted as absent since the corresponding lemmas had similarity score above optimum threshold only with EWN synsets outside the gold dataset (case 2). Only 1 of those links (2.44%) were correctly labeled when using word embeddings maximum. For verbs, 1 out of 53 (1.89%) golds links belonging case 1 were accurately predicted when using word embeddings maximum, and 0 out of 22 gold links (0%) from case 2 were predicted accurately as present when using word embeddings maximum. For adjectives, none of the gold links (0%) falling under case 1 were fixed when using word embeddings, and 4 out of 22 gold links (18.18%) belonging to case 2 were predicted as present when using word embeddings maximum. By looking at the links from case 2 that were correctly classified when using word embeddings maximum, we notice that the corresponding Jaccard similarity measures were below the optimum tuned threshold.

4.4.7 Evaluation of Fusion Models

The three fusion models described in subsection 4.3.6 are evaluated and the results are compared to the best results of each of the above link prediction approaches. Table 4.25 shows the results of overall precision, recall and F1 when the objective of the fusion model is to maximize overall F1 (equation 4.3). Using majority votes helped in improving the precision of the predictions at the expense of lower recall while a significant boost in the recall is noticed when using the similarity measures as features to train SVM. However, none of the fusion methods resulted in a better overall F1 score compared to when using MT. For nouns and verbs, using the top 5 features resulted in best F1, while for adjectives, using the top 8 features performed best. In both cases, top 5 and top 8, non-linear SVM resulted in better F1 compared to linear SVM. Moreover, using PCA did not improve the results. The top 8 similarity measures were the same across the three POS tags and their order was as follows: Direct Matching (DM) (0), DM (1), Machine Translation (MT) (1), DM (2), DM (3), MT (2), DM (4), Word Embeddings (WE) (2). We notice that none of the experiments using the EWN gloss and extended gloss made it to the top 8 as expected given the very low performance achieved when using these EWN components.

We present further analysis on the results obtained when using boosting since the behavior of the other two fusion techniques is more or less predictable. We compare the number of lemmas that did not match to any synset when using the fusion with SVM and when using direct matching with lemmatization for nouns and using raw SAMA gloss terms for verbs and adjectives. Similarly, we compare

Link Prediction Approach	Noun				Verb				Adjective			
	<i>Thr</i>	<i>Pre</i>	<i>Rec</i>	<i>F1</i>	<i>Thr</i>	<i>Pre</i>	<i>Rec</i>	<i>F1</i>	<i>Thr</i>	<i>Pre</i>	<i>Rec</i>	<i>F1</i>
Direct Matching (Exp1(noun)/Exp0)	14	15.7	50.0	23.9	0	11.2	42.7	17.7	43	22.7	29.9	25.8
Using EWN Gloss and Extended Gloss (Exp3)	22	1.4	5.4	2.2	0	2.1	43.6	4.1	13	1.0	19.4	1.9
Using MT Tables (Exp1(noun)/Exp2)	10	25.4	33.1	28.8	18	15.4	24.4	18.9	22	22.6	35.8	27.8
Using Word Embeddings (Max)	98	15.3	52.2	23.7	89	11.0	42.9	17.6	97	9.4	44.8	15.6
Fusion by Aggregation	47	24.2	31.6	27.4	37	12.5	37.0	18.7	60	20.4	31.3	24.7
Fusion by Majority Vote	8	26.3	28.0	27.1	8	13.3	31.4	18.7	8	28.8	25.4	27.0
Fusion Using Boosting	NA	8.2	63.2	14.4	NA	6.7	53.9	11.9	NA	7.8	62.7	13.8

Table 4.25: Overall performance of the best of the different explored variations for link prediction and when using fusion approaches. Pre = Precision, Rec = Recall and Thr= Threshold.

the number of lemmas from the test set that linked only to synsets outside the gold dataset. The results are shown in Fig. 4.7.

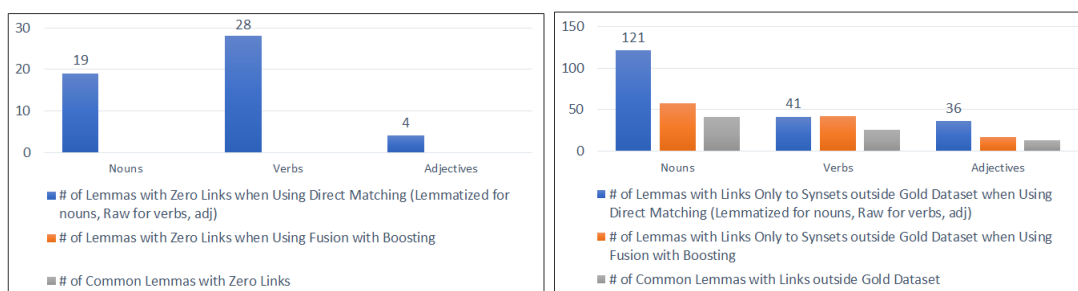


Figure 4.7: Comparison of count of lemmas with zero links and count of lemmas with links to EWN synsets outside the Gold dataset in the test set when using Fusion with boosting and best of direct matching for each POS tag.

As can be seen in Fig. 4.7, the number of lemmas with zero links decreased to 0 when using fusion with SVM resulting in highest recall for all POS tags compared to when using direct matching and any other technique. Similarly, the number of lemmas that linked only to synsets outside the gold dataset decreased for both nouns and adjectives and increased by 1 for verbs. Despite this overall decrease, the precision achieved was significantly lower for the three POS tags. In fact, many lemmas linked to synsets that are part of the gold dataset but the SAMA lemma-EWN synset pairs have as gold label “False”.

By examining specifically the random sample of 300 links from the test set, using fusion with SVM helped in predicting accurately several links with gold label “True”. For nouns and when using direct matching, 24 gold links were predicted as absent since the corresponding lemmas had a zero similarity score

with all EWN synsets (case 1). 12 of those links (50%) were correctly labeled when using fusion with boosting. 41 gold links when using direct matching were predicted as absent since the corresponding lemmas had similarity score above optimum threshold only with EWN synsets outside the gold dataset (case 2). 17 of those links (41.46%) were correctly classified when using fusion with SVM. For verbs, 15 of the 53 (28.30%) gold links belonging to case 1 were accurately predicted when using word fusion with SVM, and 6 out of 22 gold links (27.27%) from case 2 were predicted accurately as “True”. For adjectives, none of the gold links (0%) falling under case 1 were fixed when using fusion with boosting, and 11 out of 22 gold links (50.00%) belonging to case 2 were predicted as present when using fusion with SVM. The improvements observed for the specific random sample confirms the significant recall improvement achieved when using fusion with SVM. We observe that the feature vectors of the links that were classified accurately when using fusion with SVM have almost in common a value above 0.02 for the MT component. This is expected since the best performance was achieved when using MT.

4.5 Proposed Final Expansion Model

Based on the results achieved throughout the different techniques and resources explored and evaluated in the previous subsections, the following back off technique is followed to expand AWN. First, since using MT performed best, all the links that had a threshold above the optimal threshold were selected. Second, for all lemmas that did not link to any EWN synset, expansion using direct matching with SAMA gloss terms was performed. For nouns, we used raw MT English terms and lemmatized SAMA English gloss terms, while for verbs and adjectives, we utilized lemmatized MT English terms and raw SAMA English gloss terms. The choice of raw versus lemmatized was concluded based on the results achieved for each of the variation per POS tag. In total, for the nouns POS tag, 16,907 distinct lemmas linked to 18,874 EWN synsets and generated 63,247 links. For verbs, a total of 7,047 lemmas linked to 7,458 different EWN synsets and 45,527 links were generated. Last but not least, a total of 3,930 adjective lemmas were linked to 4,414 EWN synsets and as a result, 9,301 links were generated. Hence, a total of 27,884 distinct lemmas-POS were mapped to 30,746 EWN synsets and a total of 118,075 links were developed. When excluding the POS tag, a total of 26,753 lemmas is obtained. These results refer to the obtained lexicon without including the gold dataset. When the gold dataset is included, a total of 28,179 distinct lemma-POS are obtained reduced to 26,803 when POS is excluded. A total of 127,696 links is obtained. The performance on the gold dataset is shown in table . In addition to evaluating the new lexicon against the gold dataset, we randomly pick 100 links for each POS tag and we manually assess the accuracy

of the links. The accuracy obtained for nouns, verbs and adjectives is 62%, 64%, and 68% respectively. The analysis of the wrongly predicted links is congruent with the analysis of false positives of the gold test set presented in subsection 4.4.3.2. As for example, the noun SAMA lemma *muqolap* (مقلة) with SAMA gloss terms ‘eye’, ‘eyeball’ was linked to the EWN synset, 08523483, with synset terms: ‘eye’, ‘heart’, ‘center’, ‘middle’, ‘centre’, and with a gloss and extended gloss: ‘an area that is approximately central within some larger region’; ‘it is in the center of town’; ‘they ran forward into the heart of the struggle’; ‘they were in the eye of the storm’. The Arabic lemma *muqolap* corresponds to the actual biological name of the eye while the meaning of the EWN synset is metaphorical. A better Arabic lemma would have been *Eayon* (عين). As a false positive example for verbs, the lemma *>aloqaY* (ألقي) with SAMA English terms ‘arrest’, ‘deliver’, ‘place blame’, ‘throw’, was linked to the EWN synset, 01505958, with synset terms ‘arrest’, ‘get’, ‘catch’, and with a gloss and extended gloss: ‘attract and fix’, ‘his look caught her’, ‘she caught his eye’, ‘catch the attention of the waiter’. Again a metaphoric meaning of the verb *arrest* is expressed in the EWN synset. A better Arabic lemma would have been *sam~ar* (سَمَّر). The adjective lemma *lAHiq* (لاحق) with SAMA English definitions ‘afterwards’, ‘attache’, ‘joined’, ‘late’, ‘shortly’, ‘soon’, ‘subsequent’, was linked to the EWN synset, 01901186, with EWN synset terms: ‘late’, ‘belated’, ‘tardy’ and gloss: ‘after the expected or usual time’, ‘delayed’, ‘a belated birthday card’, ‘I’m late for the plane’, ‘the train is late’, ‘tardy children are sent to the principal’, ‘always tardy in making dental appointments’. The correct Arabic adjective would be *muta<ax~ir* (متأخر).

4.6 Discussion and Comparison of Expanded Lexicon with Related Work

We compare the newly developed lexicon to ArSenL on two aspects: accuracy performance on the gold dataset, and performance on subjectivity and sentiment classification.

A comparison between the performance of the proposed link prediction methods and the performance achieved when using the heuristic method followed for developing ArSenL is shown in Table 4.26. Using the test set, overall precision, recall and F1 are reported per POS tag, and an average precision, recall and F1 are computed for the three POS tags combined together for ArSenL, specifically ArSenL-Eng (the resulting lexicon from directly matching SAMA Lemmas to EWN Synsets) and for the resulting lexicon developed through our new proposed method. As can be seen in the results the of Table 4.26, the newly proposed approach improves the overall precision and

F1. Specifically, better precision and F1 are achieved for nouns, and better precision, recall and F1 are achieved for adjectives. As for verbs, a slight increase in the recall is observed at the expense of a slightly lower precision and F1. In terms of lemma counts, ArSenL includes a total of 28,325 lemmas excluding POS tags. Thus, 1,522 more lemmas than the newly proposed lexicon. For a fair comparison, we needed to exclude POS tags when comparing lemma counts. In fact, by comparing the set of lemmas in ArSenL to the set of lemmas in the new lexicon, we notice that there are 3,567 lemmas present in ArSenL but not in the new lexicon and 2,045 lemmas present in the new lexicon but not in ArSenL. By skimming through the list of lemmas only present in ArSenL, we notice that the majority include proper nouns such as kurdisotAn (کردستان), hunogAriyA (هنغاريا), himalAya (همالايا), bAfAriyA (بافاريا). These nouns were excluded in the new proposed method, specifically in the proposed set of SAMA lemmas for expansion, since they most likely do not carry a sentiment. While skimming through the lemmas that are present only in the newly developed lexicon but not in ArSenL, we notice that many new verbs and adjectives were added. We were able to retrieve links to these verbs thanks to the MT tables utilized since the SAMA gloss terms for verbs consist usually of phrases describing the meaning of the verb. The verbs and adjectives tend to bring an important semantic value unlike proper nouns. Examples of verbs include include {iToma>an~ (اطمأن) (rest assured), {ilot>am (إلتأم) (healed), and {ikotamal (إكتمل) (completed). Examples of adjectives include >aEonaf (أعنف), mutafAEil (متفاعل), and mutarAkim (متراكم).

Approach	Noun			Verb			Adjective			Average		
	Prec	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Proposed Method	20.5	42.3	27.6	12.0	31.4	17.4	23.3	46.3	31.0	17.2	38.7	23.8
ArSenL	16.6	44.0	24.1	13.3	30.1	18.4	10.8	38.8	16.9	15.2	39.0	21.9

Table 4.26: Comparison to ArSenL. Pre = Precision, Rec = Recall.

For subjectivity and sentiment classification tasks, we utilize the same corpus used for evaluating ArSenL [36]. The corpus consists of the first 400 documents from the Penn Arabic Treebank (part 1 version 3). The sentences were annotated by [77] with four labels objective, subjective-positive, subjective-negative, and subjective neutral. In addition to comparing the coverage of the two lexicons, we also evaluate the performance of the lexicons when used for subjectivity classification (objective vs. subjective) and for sentiment classification (positive vs. negative). For subjectivity classification, the subjective-positive, subjective-negative and subjective-neutral sentences were considered as subjective sentences. While for the sentiment classification task, we only consider the sentences labeled as subjective-positive and subjective-negative for evaluation. For each of the tasks, we randomly split the

dataset into 80% for training and 20% for blind testing.

We use nonlinear SVM as machine learning model with a radial basis function (rbf) as kernel. Through 5-fold crossvalidation on the training set, the kernel parameters (kernel’s width γ and regularization parameter C) were tuned to optimize F1 for the case of subjectivity classification and weighted average-F1 for the case of sentiment classification. We train the SVM classifier using sentence vectors consisting of three numerical features that reflect the sentiments expressed in each sentence, namely positivity, negativity and objectivity. The value of each feature is calculated by matching the lemmas in each sentence to each of the two lexicons separately. The corresponding scores are then accumulated and normalized by the length of the sentence. We remove all stop words in the process based on [438]. For words that occur in the lexicon multiple times, the average sentiment score is used. Three evaluations were conducted to compare the performances of ArSenL to the new lexicon. The results of the experiments are shown in Table 4.27.

		Baseline	ArSenL	ArSenL 2.0	
Coverage		NA	90.5	80.9	
Subjectivity Classification	Precision	55.2	56.7	59.6	
	Recall	100	99.4	96.5	
	F1	71.1	72.2	73.7	
Sentiment Classification	Positive	Precision	0	75.0	87.5
		Recall	0	15.3	14.3
		F1	0	25.4	24.6
	Negative	Precision	58.5	61.6	61.8
		Recall	100	96.4	98.6
		F1	73.8	75.1	76.0
	Weighted Average F1		43.2	54.5	54.6

Table 4.27: Results of subjectivity and sentiment classification.

First, we evaluate the coverage of the two lexicons. We define coverage as the percentage of lemmas (excluding stop words) covered by each lexicon. ArSenL has a higher coverage than the new lexicon. This is expected given the lemma count difference. On the other hand, the new lexicon outperformed ArSenL in both classification tasks. Using the new lexicon also outperformed the baseline results that were computed by assuming a majority vote classifier. The number of samples of the negative sentences is greater than the number of positive samples, similarly for the number of subjective sentences compared to objective ones. Although the coverage of ArSenL is greater than the new lexicon, the new lexicon performed better in both classification tasks. This performance is justified by the the better semantic quality of the new lexicon compared to ArSenL as also confirmed by the results achieved on the gold dataset.

4.7 Summary

We explored and evaluated a rich set of link prediction approaches to automatically and accurately map Arabic SAMA lemmas to EWN synsets and as a result enrich SAMA with EWN semantic and cognitive features. Sentiment scores were also extracted from SWN and we presented ArSenL 2.0, an extended version of AWN and a semantically improved version of the Arabic sentiment lexicon, ArSenL. The techniques involved using variations of SAMA English gloss terms, Machine Translation tables, EWN synset terms, glosses and extended glosses, and word embeddings. A detailed error analysis was also presented to help identify the challenges, the complexity and the issues of each of the link prediction technique in addition to their fusion. Finally, an expansion approach was proposed based on the performances of the different similarity features. The new lexicon was compared to ArSenL on different semantic aspects including the performance on the gold test dataset of known links between SAMA lemmas and EWN synsets and the performance when the sentiment lexicons are utilized in subjectivity and sentiment classification tasks. The new lexicon consistently outperformed ArSenL. The presented evaluation and analysis can be used a basis for expanding WordNets of other low resource languages.

Chapter 5

Applications

This chapter covers first a direct application of lexical resources into a real world mobile app for automatically classifying the sentiment of Arabic tweets. Then, details of context and language independent recommender systems models are presented. These models are essential components in the overall design of a decision system that integrates the opinion of the users as discussed in chapter 6.

5.1 A Light Lexicon-based Mobile Application for Sentiment Mining of Arabic Tweets

The work in this section was published in [37]. The section describes a real world mobile application that showcases the usability of ArSenL [36] for sentiment classification with a computationally inexpensive model.

5.1.1 Method Overview

The processing steps of the model are shown in Fig. 5.1. The preprocessing steps include: Tweet tokenization, hashtag removal, stemming, sentiment scores inference for the stemmed words, and then sentiment classification. The scores are then used to derive three aggregate features containing the sum of positive scores, the sum of negative scores, and the sum of objective or neutral scores. In this paper, we use objective and neutral interchangeably. These preprocessing steps are further detailed here. **Removing Hashtags:** This step is essential to clean the data from hashtags and keep their corresponding words for sentiment analysis given their importance in the sentiment of the tweet.

Stemming: Each tokenized tweet is stemmed to match it to a stemmed version of ArSenL. Lemmatization would have produced higher accuracy, however it would have required more computations. As a result, we used stemming to keep the processing light. Khojas stemmer (1999) was utilized in the implementation.

Getting the Score of Tweets: Each stemmed word is matched to the stemmed version of ArSenL in order to retrieve the corresponding sentiment scores. If a word in the tweet did not have any match in ArSenL, a zero score is assigned for each of the positive, the negative and the objective scores of the word. The sentiment scores are then summed for each tweet. It is worth noting that we tried using an average score per tweet instead of the sum but using the sum lead to better accuracy.

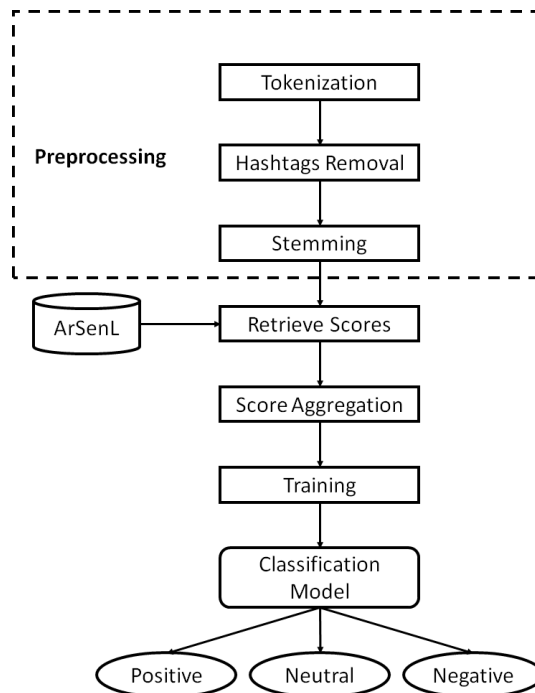


Figure 5.1: Efficient opinion mining model in Arabic for mobile use.

5.1.2 Features and Mining Model

5.1.2.1 Training Data

A corpus of 2300 [91] manually annotated Arabic Tweets (30k words) is utilized. The dataset was randomly sampled from Twitter out of 65 million unique tweets in Arabic. It was annotated by two native Arabic speakers. In case of disagreement, the two annotators discussed the issue of the tweet to resolve it. In case the disagreement remains, the tweet was dropped.

5.1.2.2 Features

The features used to build the classification model were only restricted to the sum of sentiment scores per tweet as retrieved from ArSenL. We made the features

simple in order to reduce the processing and computation efforts given that our aim is to design an energy efficient sentiment model for mobile.

5.1.2.3 Classification Model

To predict the sentiment of a tweet, we decided to use decision trees as a classification model for ease of results interpretation. The design is an ensemble classifier consisting of three binary classifiers: positive/not positive, negative/not negative and objective/not objective as shown in Fig. 5.2. In order to train each classifier, an equal number of tweets is used for each class. The results of the three classifiers are then evaluated against custom developed rules that combine the results of the three classifiers in order to assign the correct sentiment label for a given tweet: positive, negative or neutral sentiment.

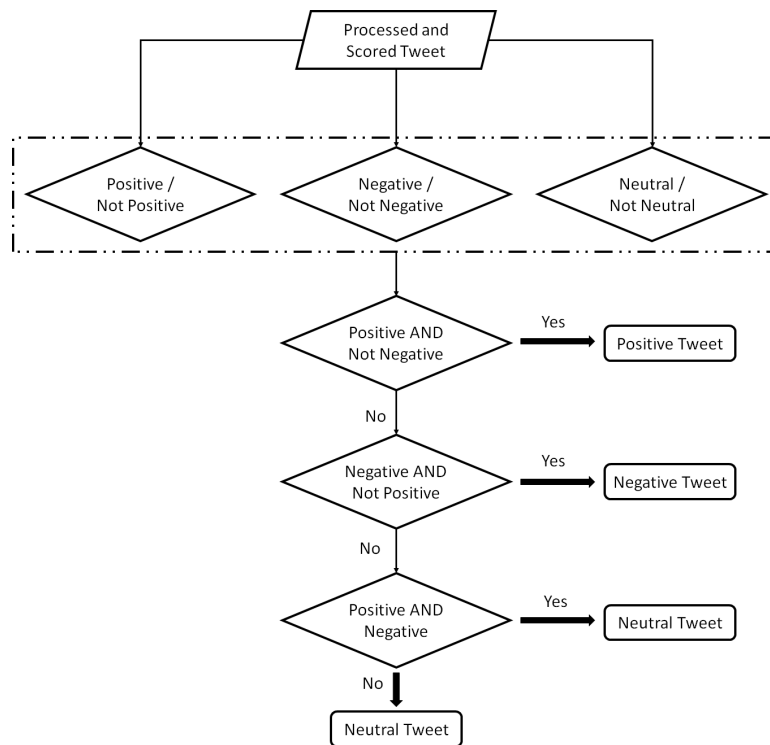


Figure 5.2: Three-way ensemble decision trees sentiment classifier.

5.1.3 Mobile Application Development

5.1.3.1 Application Architecture

A 3-tier architecture shown in Fig. 5.3 is used for the design of the application. The design is divided into three main components. The user interface is the component where the model takes as input the topic of interest and where the

tweets are displayed after being classified as positive, negative or objective. The logic part consists of the processing performed in order to match the stemmed tweets to the stemmed lemmas in ArSenL and extract sentiment scores. The sentiment scores are fed to the classification model described above. The Data component represents all the sources of data that the application makes use of: the tweets accessed through an API, filtered tweets based on the input topic, ArSenL and the classification model. No additional servers are required to perform sentiment classification. Thus, the energy is reduced since there is no need for I/O communication with a remote server or for server-level computations. The mobile application was developed for Android OS mobiles and was titled “شورأين؟” meaning ‘What is their Opinion’. It is available for download through OMA-Project website¹. An example reported in Table 5.1 to illustrate the different steps of the architecture in Figure 5.3. Below, we describe the steps involved in retrieving the sentiment of a tweet.

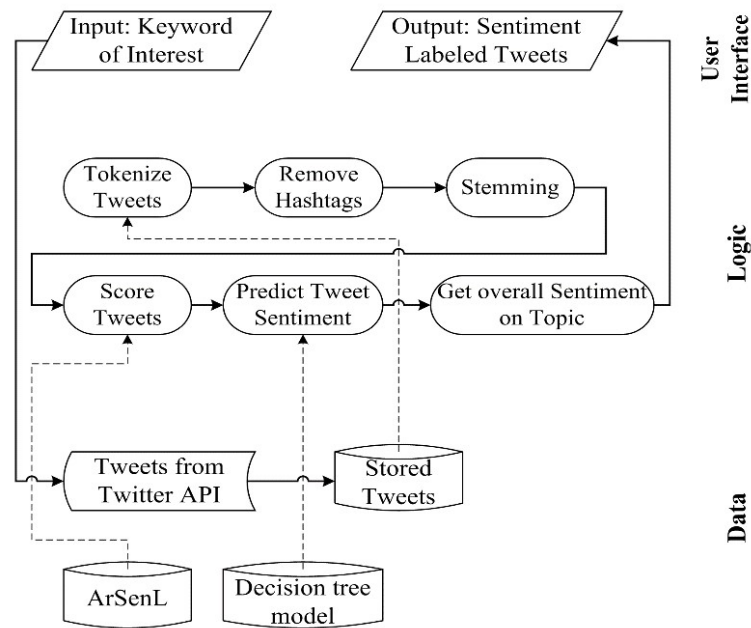


Figure 5.3: 3-tier architecture of the mobile application.

5.1.3.2 Fetching Tweets

There is a search box at the top of the main page in which the user enters the keyword of interest. Based on the keyword entered, recent tweets are fetched using Twitter API with Arabic filtering so that all fetched tweets are in Arabic. The user has the option to fetch more tweets by clicking on the Show More

¹www.oma-project.com

Tweet	لبنان بلد جميل (Lebanon is a beautiful country)		
Tokens	جميل	بلد	لبنان
Stems	جمل	بلد	لبن
Scores (Positive, Negative, Objective)	0.75; 0; 0.25	0; 0; 1	0; 0; 1
Positive/Not Positive	Positive		
Negative/Not Negative	Not Negative		
Objective/Not Objective	Objective		
Final Classification	Positive		

Table 5.1: Example of a processed tweet.

button. The fetched tweets are then stored in an array list for further processing and then deriving the related sentiment.

5.1.3.3 User Interface Design

The fetched tweets are processed and labeled as positive, negative or objective as described. The tweets are displayed to the user and colored according to their sentiment label: green color for positive sentiments, red color for negative opinions and gray color for objective tweets. Instead of looking at each tweet separately, a summary overview on the sentiments towards a specific topic can be accessed through the visual summaries available in the application. A pie chart is used to visualize the summary of the recently analyzed tweets, showing the distribution of the sentiment labels with the three colors green, red and gray. Since hashtags are essential features in tweets and are usually highly correlated with the topic of the tweet, the design of the application allows the user to see the most used hashtags corresponding to the searched topic. Another important feature in the application is the availability of the history track. This option allows the user to keep track of the evolution of sentiment distributions regarding a specific topic through time. A snapshot of the different interface options is shown in Fig. 5.4, showing classified tweets for the topic “لبنان” (Lebanon). These tweets reflect the latest tweets available on Twitter in Spring 2014.

5.1.3.4 Evaluation of the Sentiment Classification Model

As described in section 5.1.2.3, an ensemble model is used to assess the sentiment of the tweet using three decision trees. The model was developed using WEKA data mining tool. The features of the model were the sums of the three sentiment scores per tweet. The dataset which consists of 2300 manually annotated Arabic Tweets [91] (30k words) is utilized to train the model and construct the trees. The model was optimized with custom rules to achieve a high accuracy in prediction. A 5-fold cross validation was used to evaluate the developed sentiment model. Accuracy measure is used to evaluate the system. Each classifier is evaluated separately and trained using the same number of tweets per class to avoid any

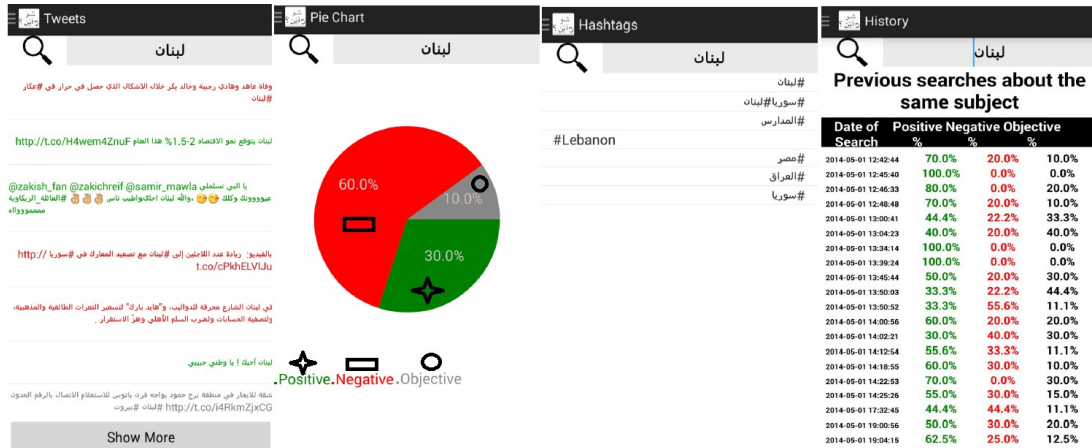


Figure 5.4: Snapshot of different user displays.

bias or over fit in the model. The results are shown in Table 5.2. An average accuracy of 67.33% was achieved for the full system.

Model	Accuracy (%)
Positive/Not Positive	61.2
Negative/Not Negative	72.9
Objective/Not Objective	67.8
Full System	67.3

Table 5.2: Accuracy percentages for each classifier and for the full system.

5.1.4 Summary

In summary, we presented in this section a light lexicon-based mobile application for sentiment mining of Arabic Tweets. A 3-tier architecture was designed to classify tweets as positive, negative or objective. The mobile application was designed to minimize energy consumption of the mobile by having an algorithm with minimal computational needs and no remote communication for computation. As an essential resource for the development of the mobile application, a stemmed version of ArSenL was generated. Different visualizations options are presented to the user. An ensemble classifier was developed based on manually annotated corpus of Arabic tweets and an average accuracy of 67.3% was achieved for sentiment classification through the mobile application.

5.2 Towards a Scalable Method for Accurate Recommender Systems Prediction

The objective of this task is to provide solutions for improved accuracy of recommender systems based on user-item rating matrix and for addressing the problem of sparsity in a user-item rating matrix using a scalable approach. Both of these challenges can be formulated as a matrix completion problem where the goal is to accurately find the missing rating values that minimize the mean absolute error. Matrix completion often seeks to find the lowest rank matrix X that matches the matrix M , which is to be recovered, for all entries in the set E of observed entries [442]. The mathematical formulation of this problem is shown in 5.1.

$$\begin{aligned} \min_X \quad & \text{rank}(X) \\ \text{subject to} \quad & X_{ij} = M_{ij} \quad \forall i, j \in E \end{aligned} \tag{5.1}$$

A popular baseline approach for solving the matrix completion problem is using collaborative filtering and evaluating similarity measure between users or items in order to estimate the missing ratings as described in the literature review section. The advantage of this approach is that it does not require additional features about items or users which in some cases may not be feasible to extract. Despite the efforts spent on enhancing the prediction accuracy of recommender systems given a user-item rating matrix, there is still room to improve the accuracy of the prediction and address the challenge of sparsity of the data using a scalable algorithm.

5.3 A Hybrid Approach for Recommender Systems

In [32], we propose a hybrid model that combines simultaneously user-based collaborative filtering and item-based collaborative filtering by adding the predicted ratings from each technique and multiplying them with a weight that incorporates the accuracy of each technique alone. The proposed approach benefits from correlation between not only users alone or items alone but from both simultaneously. Hence, the predicted result will combine two aspects of similarities: user-user similarities and item-item similarities. The rating \hat{r} for an item given a specific user is given as follows:

$$\hat{r} = \alpha \hat{r}_u + \beta \hat{r}_i \tag{5.2}$$

where, \hat{r}_u is predicted using user-based collaborative filtering and \hat{r}_i is predicted using item based collaborative filtering for the same item and user. α

and β are weights given to each of the relative ratings \hat{r}_u and \hat{r}_i . α and β are both fractions satisfying the following conditions:

$$\alpha + \beta = 1; \alpha \leq 1; \beta \leq 1 \quad (5.3)$$

The item rating \hat{r}_i can be computed using the following equation:

$$\hat{r}_i = \bar{r}_i + \frac{\sum_{j \in N(i)} sim(i, j) * (r_u, i - \bar{r}_j)}{\sum_{j \in N(i)} |sim(i, j)|} \quad (5.4)$$

where $sim(i, j)$ is the similarity between item i and item j , and $N(i)$ is the neighborhood of the item i . The user rating \hat{r}_u is computed as shown in 2.2 where $sim(u, v)$ is the similarity between users u and v , and $N(u)$ is the neighborhood of the user u . Once the ratings of the items are predicted for an active user, the Top k items are selected based on the highest ratings.

5.3.1 Choice of Weights α and β

To select the optimum values, the weights α and β have to be selected to maximize the accuracy of the recommender system. Equivalently, the choice of the weights needs to minimize the error resulting from the difference between predicted ratings and actual ratings available in training data.

While several measures are possible for assessing the accuracy of the system, we use mean absolute error (MAE) to measure the deviation of recommendations from their true user-specified values. For each rating-prediction pair $\langle p_i, q_i \rangle$, p_i being the predicted value and q_i the correct value available in the training data, the absolute error is computed as $|p_i - q_i|$. The MAE is then evaluated by examining N ratings-prediction pairs, and computing the average error as shown in the equation 5.5 below:

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \quad (5.5)$$

The lower the MAE the better the accuracy. As a result, the choice of the weights (α, β) needs to minimize the MAE as shown in 5.6:

$$\frac{argmin}{\alpha, \beta} MAE \quad (5.6)$$

To simplify the search space for (α, β) , we propose a simplified empirical approach. Based on [409], the accuracy of item-based collaborative filtering was proved to be more accurate than user-based collaborative filtering. As a result, we propose a higher weight, β , be given to the prediction performed by item-based collaborative filtering. Furthermore, since the rating scale in a recommender system consists of integers or decimals with one decimal digit, several cases are considered for the weights α & β and they are represented in Table 5.3. The

more we increase β , the more the impact of item similarities is compared to user similarities and vice versa. Fig. 5.5 shows the different MAE values obtained for the proposed α & β values. We also add a case ($\alpha = 2\beta$, or $\alpha = 2/3$ & $\beta = 1/3$) to make sure that item-based collaborative filtering is indeed more accurate than user-based collaborative filtering. It is worth noting that the two special cases: ($\alpha = 1, \beta = 0$) and ($\alpha = 0, \beta = 1$) correspond to using user-based collaborative filtering alone and item-based collaborative filtering alone respectively.

Case	α	β
$\beta = \alpha$	1/2	1/2
$\beta = 2 * \alpha$	1/3	2/3
$\beta = 3 * \alpha$	1/4	3/4
$\beta = 4 * \alpha$	1/5	4/5
$\beta = 5 * \alpha$	1/6	5/6
$\beta = 6 * \alpha$	1/7	6/7
$\beta = 7 * \alpha$	1/8	7/8
$\alpha = 2 * \beta$	2/3	1/3

Table 5.3: Values of α and β used for testing the proposed method.

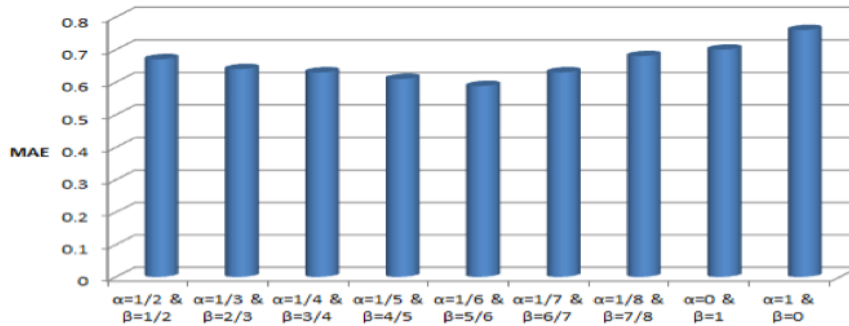


Figure 5.5: Simulation results for different values of α and β .

The MAE for each case is computed by making use of different combination of data sets and the optimal weights will be the ones corresponding to the case with the lowest MAE as illustrated in equation 5.6. Moreover, in order to improve the time performance of the system, a fixed neighborhood size N is set. The highest N similarity values are selected for each technique, i.e., the user-user similarity measure and the item-item similarity measure. Since the closest users and items are expected to have the biggest impact on accuracy, the impact of choosing only N neighbors to perform the calculation of the ratings to be predicted is expected to be negligible on accuracy compared to the gain in time performance.

5.3.2 Evaluation

In order to evaluate the proposed system, experiments are conducted on data selected from MovieLens², a web-based research recommender system that debuted in 1997. The data was collected from hundreds of users who visit MovieLens to rate and receive recommendations for movies. Several data sets exist on the site, and the 100k ratings was used for the evaluation. The selection of this dataset specifically is made in order to compare our results to user-based collaborative filtering and item-based collaborative filtering performed on the same dataset as described in [409].

The data is stored in text files that we transformed to a user-item matrix. The data is divided into 5 training sets and 5 corresponding testing sets and thus, a 5-fold cross validation approach was applied (i.e. 80% training data and 20% test data) to evaluate our system. The accuracy of the proposed technique was compared to user-based collaborative filtering as stand-alone and item-based collaborative filtering as stand-alone.

We search for the best weights following the proposed values of α and β using an empirical approach by observing the MAE for the different combinations of α and β as described in the previous section and listed in Table 5.3. It was observed that $\alpha = 1/6$ and $\beta = 5/6$ produced the lowest MAE compared to the other suggested combinations as depicted in Fig. 5.5. Although the combination $\alpha = 1/8$ and $\beta = 7/8$ was expected to represent the optimum solution since the weight accorded for item-based collaborative filtering is higher. This behavior can be explained by observing that the relatively reduced α factor hides the intrinsic similarities and relations that can be extracted among users through user-based collaborative filtering. Thus, $\alpha = 1/6$ and $\beta = 5/6$ were the optimal coefficients as found through empirical analysis.

To test the proposed technique, a neighborhood size of $N = 20$ was used based on [409] where a neighborhood size of 20 was optimal in terms of MAE and performance. For larger neighborhood sizes, no significant improvement was obtained in terms of MAE. The simulation was performed in MATLAB on a Windows 7 with Intel I7 2.4GHz as CPU and 6GB RAM. The results are shown in Fig. 5.5 where for $\alpha = 1/6$ and $\beta = 5/6$ we obtain the lowest MAE compared to the other combinations and hence, this is the optimal solution. In order to compare our proposed approach to state-of-the-art techniques, we select the optimum evaluated combination and compare the resulting MAE to the ones measured by using user-based collaborative filtering and item-based collaborative filtering separately. As shown in Fig. 5.6, the proposed technique gives a better accuracy with an improvement of 23% over user-based collaborative filtering and 16% over item-based collaborative filtering.

²<http://www.grouplens.org/node/73>

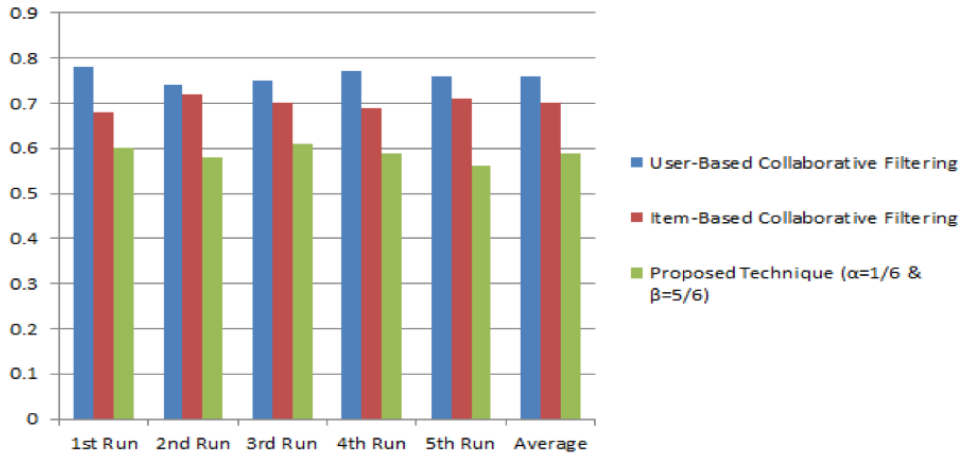


Figure 5.6: Simulation results for user-based, item-based and our proposed hybrid-base collaborative filtering. The y-axis represents the MAE.

5.3.3 Summary

In this section, we proposed a new hybrid method for recommender systems based on simultaneous combination of user-based and item-based collaborative filtering. The results showed improvements in accuracy compared to using user-based or item-based collaborative filtering separately. Moreover, the proposed technique addresses two common challenges of recommender systems, namely sparsity of data and improved accuracy of recommender system by combining the hidden relations between users and items.

5.4 Multiresolution Approach for Recommender Systems

This section describes the work that was published in [34, 33].

5.4.1 Methodology

5.4.1.1 Overview

The objective of this method is to further improve the accuracy of recommender systems for sparse user-item matrix using a scalable approach. User-Item Rating matrices are known to be sparse [430]: thousands to millions of users and items and there are much more missing ratings than available ones. Since the objective is to improve accuracy given a user-item rating matrix only, one would need a method that would harvest all possible knowledge from the matrix. For this purpose, mathematical theories are exploited about matrix

completion as described in [443]. Coifman and Gavish in [443] proposes a harmonic analysis approach that finds the interplay between the columns and the rows of a matrix at different granularities. In brief, their approach can be thought of to be similar to an image denoising problem: transforming the image into a new dimensional space, extracting the dominant coefficients and inverse transform the image to the original space.

To address the sparsity of the data in this objective, we propose a multiresolution approach for matrix completion. An overview of the steps showing how this approach is applied to the user-item rating matrix is represented in Fig. 5.7.

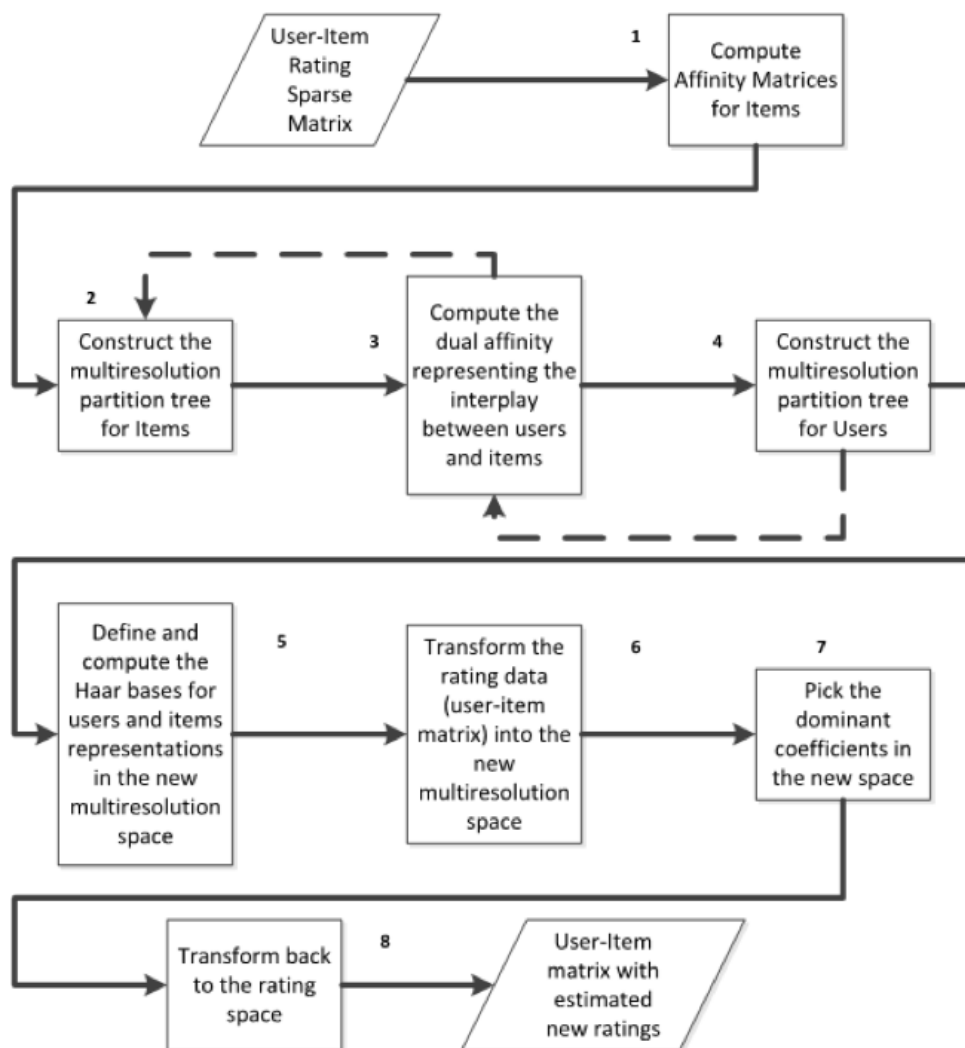


Figure 5.7: Multiresolution approach for matrix completion.

In step 1, the similarities among users and items are computed separately using correlation measures. The similarity measures are represented by so called

affinity matrices. For user affinity, similarity is measured between rows of the user-item matrix. For item affinity, similarity is measured between columns. Steps 2-4 constitute an iterative process that converges to two multiresolution representations of the user-item matrix. The process initially starts by deriving a multiresolution partition tree for items by clustering similar groups of items (columns) together at different granularity levels. In step 3, the interplay and similarity between users (rows) and items (columns) are measured by computing similarity between users (rows) based on the similarities across rows of the items partition tree, i.e. rows across the multiresolution levels of the items partition tree. This measure of interplay is called dual affinity. Similarly, the interplay and similarity between items (columns) and users (rows) are measured by computing similarity between items (columns) based on the similarities across columns of the users partition tree derived in step 4, i.e. columns across the multiresolution levels of the users partition tree. iterative approach in steps 2-4 is repeated until the partition trees converge with very little change from one iteration to the next. The results of this convergence are two multiresolution partition trees capturing the interplay: one for the users and one for the items. Through steps 5 and 6, the user-item matrix is transformed to the new multiresolution space with the use of the orthonormal Haar-like bases constructed from the partition trees. The product space spanned by the tensor product of the Haar-like bases can then be used to represent the original rating matrix in the new space. The orthonormal representation is constructed to represent the original user-item matrix in the new space of the multiresolution partition trees. In step 7, similar to a Wavelet transform, this new transform can be used to efficiently compress and denoise the user-item matrix. As a result, in step 7 of the approach, the dominating coefficients in the transformed space are selected to provide the efficient representation. Step 8 involves a step similar to an inverse wavelet transform. The dominant coefficients selected from the previous step are transformed back to the original user-item space. Typically, a small percentage of the coefficients are used to reconstruct the new estimated user-item matrix with the desired capture for previously missing ratings. In the following sections we describe each step mathematically.

5.4.1.2 User-user and Item-item Affinity Matrices

Given a user-item rating matrix M of size $m * n$, m represents the number of users, and n represents the number of items. Let $X = M$, and $Y = M^T$. These two matrices X and Y are used in formulating item related affinity matrix W_X and user related affinity matrix W_Y . We define the item affinity matrix W_X as follows in 5.7 using similarity measures for all items i and j .

$$W_X(i, j) = \frac{\sum_{u \in X_{ij}} (r_{u,i} - \bar{r}_i) * (r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in X_{ij}} (r_{u,i} - \bar{r}_i)^2 * \sum_{u \in X_{ij}} (r_{u,j} - \bar{r}_j)^2}} \quad (5.7)$$

where X_{ij} represents all users who rated both items i and j , where $i \neq j$, and $1 < i, j < n$. n is the total number of items. $r_{u,i}$ and $r_{u,j}$ are the ratings provided by user u to items i and j respectively. \bar{r}_i and \bar{r}_j are the averages of all the ratings provided for items i and j respectively. Similarly, one can compute the affinity matrix W_Y among users using 5.8.

$$W_Y(u, v) = \frac{\sum_{i \in Y_{uv}} (r_{u,i} - \bar{r}_u) * (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in Y_{uv}} (r_{u,i} - \bar{r}_u)^2 * \sum_{i \in Y_{uv}} (r_{v,i} - \bar{r}_v)^2}} \quad (5.8)$$

where Y_{uv} is the set of items that were simultaneously rated by users u and v . $r_{u,i}$ and $r_{v,i}$ are the ratings provided for item i by users u and v respectively, and where $u \neq v$, $1 < u, v < m$. m is the total number of users. \bar{r}_u and \bar{r}_v are the averages of all the ratings provided by users u and v respectively. The computations of affinity can also be performed using other statistical measures indicating distance or similarity such as cosine measures or Pearson correlation. We use correlation for ease of illustration.

5.4.1.3 Multiresolution Partition Trees

Given a user-item matrix M , one can organize X , the set of items ratings (columns), and Y , the set of users ratings (rows) into two partition trees using the affinity matrices W_X and W_Y evaluated for the items and users separately. These partition trees, T_X and T_Y , give a hierarchical multiresolution grouping, called folders, of similar users and items respectively.

To illustrate the approach for partition trees of X and Y , consider the partition tree T_X for items X . Every node in the tree is a folder, and represents a cluster of items at the level of the node in the tree. A sample partition tree of three levels ($L = 3$) is shown in Fig. 5.8. The finest level corresponds to the leafs of the partition tree. At this finest level, the partition is composed of folders where each folder represents one column from the user-item matrix, i.e. a set of users ratings for an item. At higher levels of the tree, starting from the finest level (leafs), the folders from the previous level are grouped to form a coarser partition at the next level up the tree. For a tree of depth L , X_l represents the set of nodes at any level, where $1 < l < L$. X_1 ($l = 1$) represents the root of the tree, and X_L represents the set of nodes at the finest level. The full partition tree T_X is represented by X_1, X_L . For each level l , where $1 \leq l \leq L$, the partition is composed of $n(l)$ mutually disjoint folders X_i^l where $1 \leq i \leq n(l)$.

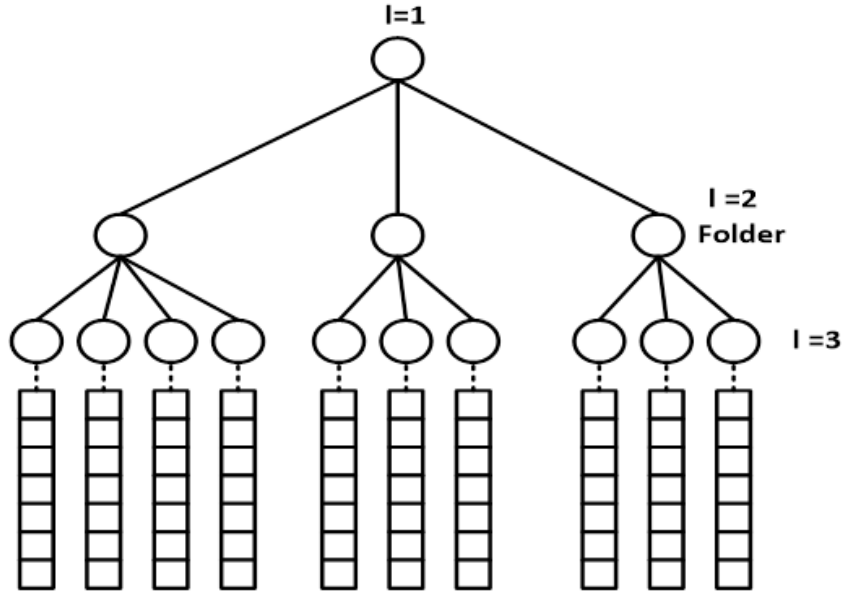


Figure 5.8: Sample partition tree with three levels ($L = 3$) for items showing the groups of items' folders as nodes at multiple resolutions. At the finest level ($l = 3$), each node corresponds to a column in the user-item matrix.

The algorithm for the construction of the partition tree is shown in algorithm 1. At the finest level, each node contains the set of elements in a column from the item matrix X , and the set of nodes is equal to the number of columns. To get coarser representations in the tree, the clustering is done bottom-up. Starting from the finest level, nodes at the finer level are clustered to form the next level up in the tree. A notion of distance between nodes needs to be defined in order to group nodes. We choose the diffusion distance as defined in [443] since it was shown to help capture the overall geometry of the data for multi-scale analysis. The diffusion distance provides means of computing distances between data in a transformed space, similar to other transforms such as Principal Component Analysis (PCA). As a result, the diffusion distances can be computed by finding the distances between vectors in a new space defined by the eigenvectors of the affinity matrix W_X .

Specifically, a diffusion distance can be computed as follows: first, the affinity matrix W_X is normalized such that the sum of the elements in each row of W_X is equal to 1 as shown in 5.9 and 5.10. O is a diagonal matrix.

$$P_X = O^{-1} * W_X \quad (5.9)$$

$$O_{i,i} = \sum_j W_X(i,j) \quad (5.10)$$

ALGORITHM 1: Creating partition trees on items.

Data:

X : Set of item columns from the user-item matrix

W_X : Affinity matrix corresponding to items

k_n : Number of nearest neighbors used to compute average diffusion distance

Output:

T_X : partition tree for items;

```
/*Initial Finest Level*/
 $D_X = \text{computeDiffusionDistance}(W_X)$ ;
 $Rho = \text{computeBaseRadius}(X, D_X, k_n)$ ;
 $\text{InitialSetofCentroids} = \text{findCentroids}(D_X, X, Rho)$ ;
 $count = 1$ ;
for each  $centroid$  in  $\text{InitialSetofCentroids}$  do
     $\text{Folder} = \text{groupItems}(X, D_X, Rho, centroid)$ ;
    /*A folder is represented by its centroid*/
     $\text{Folders}[\text{level}+1, count] = \text{Folder}$ ;
     $count ++$ ;
end
/* Moving from a finer level to coarser levels in the partition tree:*/
 $number\_of\_Folders\_in\_Level = count$ ;
while  $number\_of\_Folders\_in\_Level \neq 1$  do
     $level ++$ ;
     $k_n = \text{ceil}(k_n/2)$ ;
     $Rho = \text{computeBaseRadius}(X, D_X, k_n)$ ;
     $W_{Temp} = \text{computeAffinityBetweenFolders}(\text{Folders}[\text{level}, :])$ ;
     $D_{Temp} = \text{computeDiffusionDistance}(W_{Temp})$ ;
     $\text{TempCentroids} = \text{findCentroids}(D_{Temp}, \text{Folders}[\text{level}, :], Rho)$ ;
     $count = 1$ ;
    for  $centroid$  in  $\text{TempCentroids}$  do
         $\text{Folder} = \text{groupItems}(\text{Folders}[\text{level}, :], D_{Temp}, Rho, centroid)$ ;
         $\text{Folders}[\text{level}+1, count] = \text{Folder}$ ;  $count ++$ ;
    end
     $number\_of\_Folders\_in\_Level = count$ ;
end
 $T_X = \text{Folders}$ ;
```

Then, we compute the eigenvectors and eigenvalues of P_X^T . The diffusion distance between two items is computed by finding the L_2 norm between the corresponding eigenvectors to these items using 5.11.

$$D_X(i, j) = \|A_X(i) - A_X(j)\| \quad (5.11)$$

where $A_X(i)$ represents the eigenvectors of item i for the normalized affinity matrix P_{X^T} , and similarly for item j . A more detailed explanation of the diffusion map theory can be found in [444].

Clustering starts at the finest level L by computing a base radius $\rho > 0$, by averaging the diffusion distance between an item and its KNN neighbors. The choice of number of nearest neighbors, k_n , in the KNN method is manually set at the first iteration and is then decreased by half when moving to the next level. The choice can be experimentally chosen to achieve highest accuracy from a given set of training data. Once the radius ρ is computed, the items data are clustered into disjoint groups of radius ρ , called balls. The number of resulting balls is denoted by k , and the centers of the balls, or centroids, are used to represent the disjoint folders at the next level of the tree. At the finest level, a node i , which corresponds to an item data, is merged with a group of centroid z_i and belongs to the corresponding folder X_i^{L-1} as per 5.12:

$$X_i^{L-1} = \{i \in X | D_X(i, z_i) < \rho\} \quad (5.12)$$

for $i = 1 \dots n(L-1)$, where $n(L-1) = \#z_{L-1}$ is the number of centroids at level $L-1$. In order to move to the next level $L-2$ of the partition tree, we define an affinity between folders at nodes i and j , as in 5.13:

$$\hat{W}_X(i, j) = \langle W_{X_i^{L-1}}, W_{X_j^{L-1}} \rangle = \sum_{x \in X_i^{L-1}} \sum_{y \in X_j^{L-1}} \sum_r W_X(x, r) W_X(r, y) \quad (5.13)$$

$\sum_r W_X(x, r) W_X(r, y)$ is an affinity measure for reaching node x to node y in two steps. This hierarchical grouping process is repeated until the coarsest level ($l=1$) is reached, with a single folder X^l .

5.4.1.4 Dual Affinity between Users and Items

The partition tree of the items (columns), T_X , is then used to define the interplay between items and users through a dual affinity matrix. This dual affinity is also called a dual geometry on the users (rows) of M . The dual affinity on users can be computed by taking the similarity between users at every level of the item partition tree. A similar approach can be done for dual affinity on items. To represent the average of similarities across different levels of the partition tree, consider the case of dual affinity on users. The idea is to compute similarity in rows of the original user-item matrix M , by computing the similarity of the

corresponding rows across the multiresolution tree achieved on items. The novelty in this approach is that it captures an overall geometry of the users that depends on the geometry of the items and vice versa. Because of these dependencies, we refer to these geometries as a coupled geometry. Interpreting users ratings as functions or mappings on T_X is similar to many wavelet-based norms (e.g. Besov norms).

First, we define a mapping from a row in the original matrix to a row at any level of the partition tree. At the finest level, the elements of a row y from the original matrix M correspond to the cells $M(y, i)$ corresponding to item i . This mapping process can be generalized, as shown in 5.14, to identify cells at any levels of the partition tree, where M_l represents the user-item matrix at level l , and $l \leq i \leq n(l)$. The set of values for row y can be represented by $M_l(y)$. With this mapping in place, we can define dual affinity between rows u and v based on the partition T_X as the average of similarities between the rows at every level as shown in equation 5.15 and illustrated in Fig. 5.9.

$$\text{For each } y \in M : y \mapsto M_l(y, i) \quad (5.14)$$

$$W_{T_X}(u, v) = \frac{1}{L} \sum_{l=1}^L W(M_l(u), M_l(v)) \quad (5.15)$$

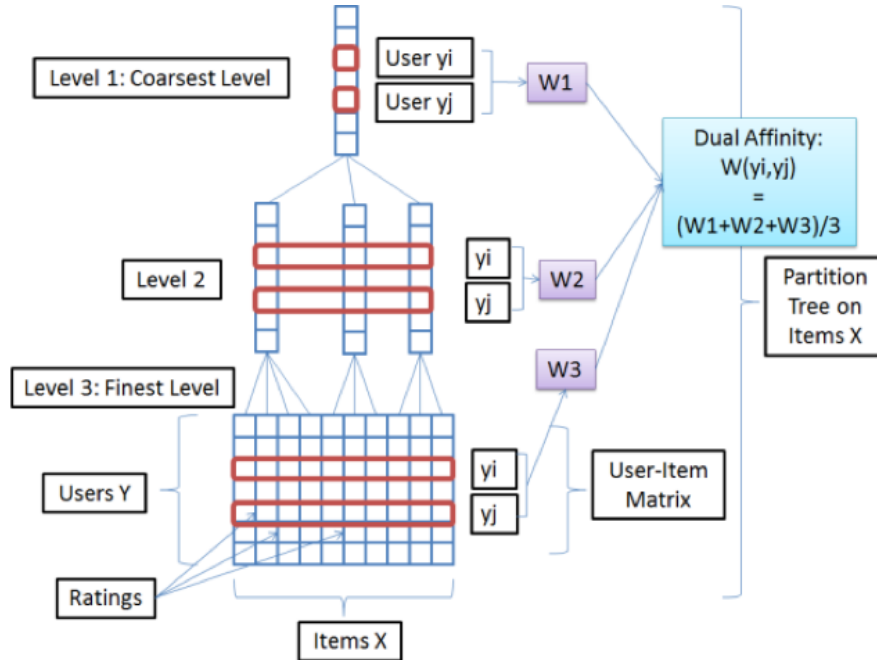


Figure 5.9: Example of using dual affinity from items partition tree to compute similarity for users.

The example in Fig. 5.9 shows that there are three available columns at level 2 of the tree. The dual affinity is the average of the three affinities computed at each level respectively as shown in Fig. 5.9. Similarly, the dual affinity on items based on partition T_Y is defined as in 5.16, where T_Y is the partition tree for users constructed based on the user affinity matrix W_Y , M_l^T represents the transpose of the original user item matrix at level l fo the partition tree T_Y . We iterate the computations of the dual affinities on X & Y until 5.17 and 5.18 are satisfied, where W_X^k represents the updated affinity matrix at iteration k . Ideally, the difference between iterations should converge to 0. However, the ideal equality situation may not exist, or at least may take a long time to achieve. As a result, the choice of ϵ provides a trade-off between best possible accuracy and computation cost. The closer the value of ϵ is to 0, the more the time the algorithm takes for convergence. The results of these iterations give us the desired two multiresolution partition trees, one for the items and one for the users.

$$W_{T_Y}(i, j) = \frac{1}{L} \sum_{l=1}^L W(M_l^T(i), M_l^T(j)) \quad (5.16)$$

$$\|W_X^{k+1} - W_X^k\| < \epsilon \quad (5.17)$$

$$\|W_Y^{k+1} - W_Y^k\| < \epsilon \quad (5.18)$$

5.4.1.5 Transformation of User-Item Matrix into Multiresolution Space

Once the partition trees are built, we construct an overall representation of the whole tree, which would allow us to project the original matrix into a multiresolution space represented by the two trees of users and items. This is equivalent to a scale-space transformation in two dimensions X and Y . The items tree provides the equivalent of the X-transformation, and the users tree provides the equivalent of the Y-transformation. This can be represented mathematically by the X and Y transformations provided by two matrices H_X and H_Y as shown in equation 5.19, where M_{Coeff} represents the coefficients of the user-item matrix in the multiresolution space. Specifically, let the basis derived from the partition tree T_X be represented by an $n * n$ matrix H_X , where the columns are the basis vectors. The rows of M are first projected to have coordinates in H_X . The columns of M are then projected to the Haar basis matrix H_Y derived from the partition tree T_Y . As a result, M can be represented using the tensor product of the Haar bases H_X and H_Y , and one can get the coefficients as in 5.19.

$$M_{Coeff} = H_Y * M * H_X^T \quad (5.19)$$

The two matrices H_X and H_Y can be derived by a set of Haar bases for the users tree and items tree respectively. The construction is performed by adding the orthonormal bases constructed at each level of the partition tree using a Gram-Schmidt process, starting with the coarsest level (level 1) and iterating to finer levels until level L . Intuitively, the construction is equivalent to having low-pass filters to provide the projection to coarsest level of a tree, and high-pass filters for the details in the finer levels. In fact, the basis is derived by first projecting into the coarsest level. Subsequently, the projections at a finer level of the tree are subtracted from projections at the previous coarser level, to provide the components of the bases for the finer level. To illustrate of the construction of the bases, lets consider the example shown in Fig. 5.9. At the coarsest level, the folder was generated by consecutive clustering as was explained in subsection 5.4.1.3. As a result, the folder at the coarsest level corresponds to 10 columns (one for each item). Since there is only one folder at this coarsest level, the projection is simply a scalar. Since an orthonormal basis is needed, the value of the Haar basis is of dimension 1 and its value is $1/\sqrt{10}$. For level 2, there are 3 folders. Thus a projection into these three vectors would require a basis matrix of size $3 * 3$, call it B as in equation 5.20, where e_1 , e_2 , and e_3 represent the orthonormal basis vectors for level 2. One of the vectors in this new basis B , e_1 can be extended from the basis of the finer level. Thus, the three cells of the first column vector e_1 of B have the value of $1/\sqrt{10}$.

$$B = [e_1 \quad e_2 \quad e_3] \quad (5.20)$$

Let f_1 , f_2 and f_3 denote the folders from left to right. Let s_1 , s_2 and s_3 denote the folder size of f_1 , f_2 and f_3 respectively. In this case, $s_1 = 4$ and $s_2 = s_3 = 3$. To find the remaining two column vectors of B a Gram-Schmidt process is adopted. The steps are as follows: first, create a diagonal matrix $Diag$ of size $3 * 3$ whose diagonal entries are n_1 , n_2 and n_3 , and which correspond to the sizes of the three folders at level $l = 2$ respectively. Then, define a matrix A of size $3 * 2$ as shown in 5.21. Let v_2 and v_3 denote the first and second column of A . Subsequently, the Gram-Schmidt process is applied on v_2 and v_3 to derive the orthonormal basis as shown in 5.22, 5.23 and 5.24. At level $l = 3$, this process is repeated to generate a $10 * 10$ Haar basis matrix. Since this level is the finest level, the resulting matrix is the desired H_X Haar basis matrix. w_3 is residual in v_3 after subtracting the projection on e_2 . This leads to three orthogonal basis v_1 , v_2 , and w_3 . v_2 and w_3 are further normalized to provide the three orthonormal bases e_1 , e_2 , and e_3 .

$$A = \begin{pmatrix} 1/n_1 & 0 \\ -1/n_2 & 1/n_2 \\ 0 & -1/n_3 \end{pmatrix} \quad (5.21)$$

$$e_2 = \frac{v_2}{\sqrt{v_2^T * Diag * v_2}} \quad (5.22)$$

$$w_3 = v_3 - \frac{v_3^T * Diag * e_2}{e_2^T * Diag * e_2} e_2 \quad (5.23)$$

$$e_3 = \frac{w_3}{\sqrt{w_3^T * Diag * w_3}} \quad (5.24)$$

5.4.1.6 Inverse Transform to Estimate Missing Ratings

After computing the coefficients as in 5.19, we select the dominant coefficients by keeping the ones that are greater than a pre-defined threshold R and eliminating other values as per 5.25, where “ $*$ ” is the pointwise multiplication. The size of the threshold is directly linked to the size of the support of the tensor product of the two Haar-like basis functions.

$$M_{Coeff} = M_{Coeff} * \text{boolean}(M_{Coeff} > R) \quad (5.25)$$

Extending on the theory of Coifman et al. [443], one only needs to consider the coefficients that are large enough in order to reconstruct an updated filled user-item matrix with estimation of missed ratings. By this method, we would only keep the relevant coefficients that will be involved in approximating and filling the original user-item matrix. The reconstruction is performed by following 5.26 where \hat{M} is the final enhanced user-item matrix with the desired estimated ratings.

$$\hat{M} = H_Y^T * M_{Coeff} * H_X \quad (5.26)$$

5.4.2 Evaluation

Several experiments are conducted to evaluate the effectiveness of the proposed method. In 5.4.2.1, we compare the accuracy of our approach against conventional user-based and item-based collaborative filtering techniques. Then in 5.4.2.2, we compare the accuracy of our approach against state-of-the-art methods using larger data. In 5.4.2.3, we study the time performance of the multiresolution approach and compare it to recent and state-of-the-art methods. In 3.6.6.4, we run the proposed approach with Netflix data set.

5.4.2.1 Comparison to Conventional Methods

The experiments described in this section are conducted on data selected from MovieLens³, a web-based movie recommendation system that debuted in 1997.

³<http://www.grouplens.org/node/73>

The selection of this dataset is made specifically in order to compare our results to user-based collaborative filtering and item-based collaborative filtering performed on the same dataset as described in [409]). The data was collected from hundreds of users who had visited MovieLens to rate and receive recommendations for movies. Several data sets exist on the site. For the chosen data set, 100,000 ratings provided by 943 users to 1682 items were used for the evaluation. The data has a sparsity level of 0.94, which indicates that the matrix has 94% of its entries equal to zero and a high degree of sparsity. The sparsity level is computed as in 5.27.

$$Sparsity\ Level = 1 - \frac{nonzero\ entries}{Total\ entries} \quad (5.27)$$

The data was divided into 5 training sets and 5 corresponding testing sets and thus, a 5-fold cross validation approach was applied (i.e. 80% training data and 20% test data). The accuracy of the proposed technique was compared to user-based collaborative filtering as stand-alone and item-based collaborative filtering as stand-alone. For fair comparisons, the algorithm was repeated 5 times, and the results were averaged consistent with the spin cycle procedure. While several measures are possible for assessing the accuracy of the system, we used mean absolute error (MAE) 5.5 to measure the deviation of recommendations from their true user-specified values. For each rating-prediction pair $\langle p_i, q_i \rangle$, p_i being the predicted value and q_i the correct value available in the testing data, the absolute error is computed as $|p_i - q_i|$. The MAE is then evaluated by examining all N ratings-prediction pairs. The lower the MAE, the better the estimation is of missing ratings. Fig. 5.10 shows the results achieved in terms of MAE compared to user-based collaborative filtering and item-based one for each of the five runs. As can be seen in Fig. 5.10, the proposed Harmonic Analysis approach achieved an improvement of 40% compared to user-based collaborative filtering and item-based collaborative filtering. The harmonic analysis approach has the lowest MAE compared to the two other methods.

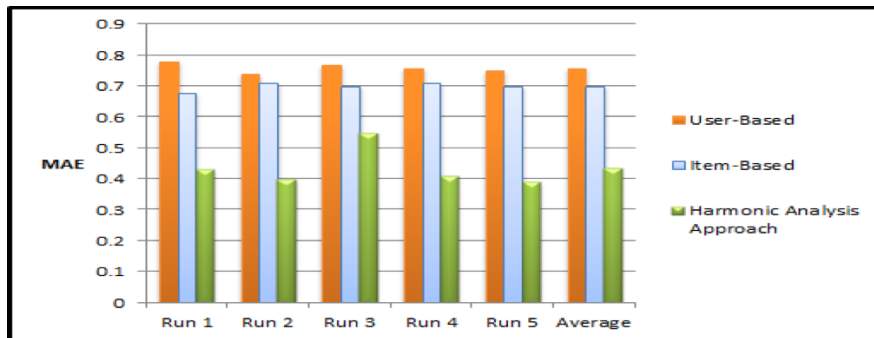


Figure 5.10: Mean Absolute Error for user-based and item-based collaborative filtering and for Harmonic Analysis approach.

Method	1M ratings	10M ratings
Multiresolution Proposed Approach	0.65	0.61
Mixed Matrix Factorization [445]	0.86	0.84
Fuzzy Clustering [446]	0.72	0.71
SVD [423]	0.73	0.72

Table 5.4: MAE for 1M and 10M MovieLens datasets.

5.4.2.2 Comparison with State-of-the-Art Methods

This set of experiments are targeted for comparing proposed approach with some more recent and state of the art work in the field [445, 446, 423]. The authors in [445] present a mixed matrix factorization approach that relies on exploiting latent factors and extracting the context of the user to predict item ratings. Treerattanapittak et al. propose in [446] an approach for improving collaborative filtering using a fuzzy clustering algorithm. For the experiments, we choose two different large datasets: the MovieLens 1M ratings which includes 6040 users and 3952 movies, and the MovieLens 10M ratings which has 10681 movies and 71567 users. A portion of the user partition tree the 1M dataset is illustrated with 58 levels in Fig. 5.11. Following the same testing process described in the previous section, the results are reported in Table 5.4. The new method showed, on average, an accuracy improvement of around 25%, 13% and 14% compared to [445, 446, 423] respectively. It is important to note that our approach as well as the one proposed in [446] are based on the provided ratings only whereas the one in [445] investigates latent factors, context and other inputs that are required to be provided by the user. This could be a disadvantage since many users prefer to skip the process of providing additional information.

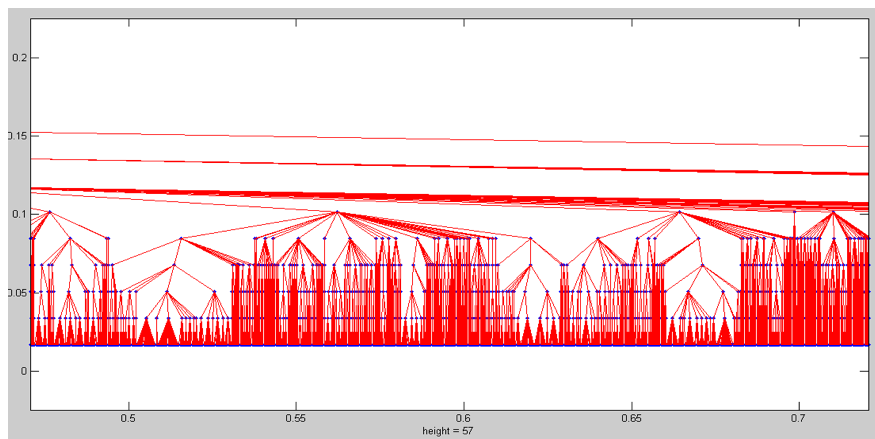


Figure 5.11: Portion of the partition tree on users of the 1M dataset.

5.4.2.3 Analysis of Performance

To evaluate performance of the algorithm, we evaluated the trade-off between system accuracy and time of running the algorithm. The experiments were implemented on MATLAB installed on an Intel core I7 device with 6GB DDR3 RAM running Windows 7 64-bit. Time measurements were collect to reflect the duration needed to run one full iteration of the algorithm and reconstruct the user-item matrix. 5 iterations were performed and ϵ was set to 10^{-4} . As shown in Fig. 5.12, the experiments indicated that the total algorithm time decreases with the number of nearest neighbors k_n used to construct the partition trees. When k_n decreases to less than 5, the algorithm time increases exponentially.

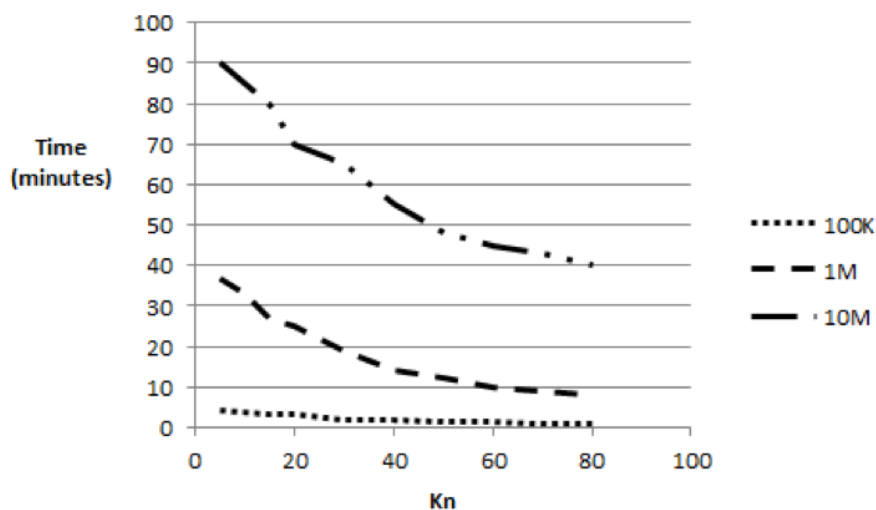


Figure 5.12: System Performance (in minutes) for running the proposed approach in terms of k_n for the three cases of Movielens dataset: 100 K, 1M, and 10M.

In Fig. 5.13, the MAE and time measurements are plotted based on varying the k_n parameter for the 10M dataset. It can be seen from this graph that the choice of k_n gives a trade-off between accuracy and time. As can be seen in the graph, MAE increases when k_n increases while the time required by the algorithm decreases when k_n increases. By checking the corresponding accuracies, the k_n values were chosen to be 15, 40 and 70 for the datasets 100k, 1M and 10M ratings respectively. These choices of k_n were based on the variation of time compared to the variation of MAE for each case of k_n . We chose a point where the MAE gave a higher trade-off of accuracy versus computation time. As an example for the 10M dataset, the choice is pointed out by the arrow in Fig. 5.13.

For comparison of time performance, we provide time measurements per iteration for the 1M dataset in comparison with state of the art approaches as shown in Table 5.5. As seen in Table 5.5, the proposed approach provides better scalability performance. It is worth noting that the time comparison per

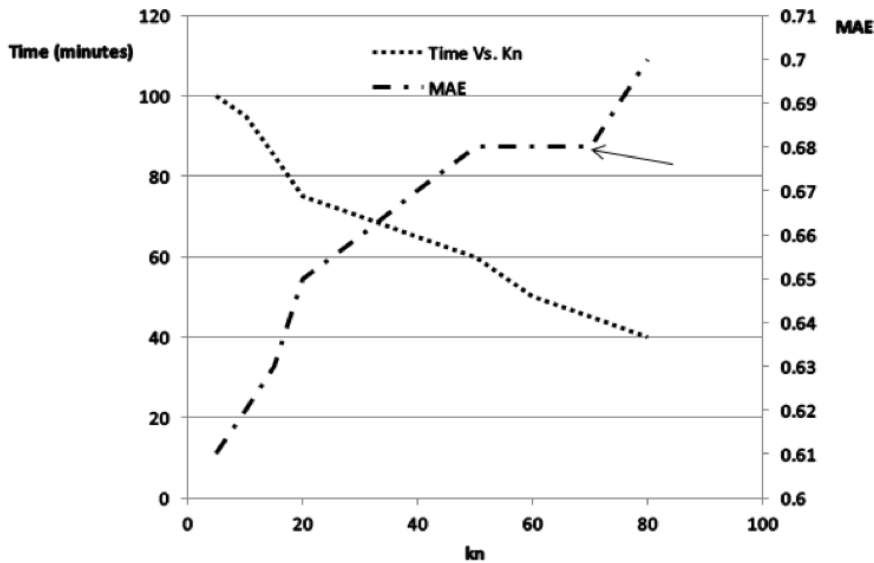


Figure 5.13: Performance (in minutes) and MAE versus choice of nearest neighbors (k_n) parameter for the 10M MovieLens dataset.

Method	Time in minutes per iteration
Multiresolution Proposed Approach	16
Mixed Matrix Factorization [445]	19
Fuzzy Clustering [446]	21
SVD [423]	18

Table 5.5: Comparison of time performance per iteration for the 1M MovieLens Dataset

iteration was chosen for consistency and fairness in comparison. Since the other methods are also iterative and the accuracy of their system also improves with the increase in the number of iterations, we provided time comparison per iteration. This reporting is also consistent with the way previous results were reported.

5.4.2.4 Using Netflix Dataset

In this section, we run the proposed approach on the Netflix dataset⁴. The Netflix dataset consists of more than 100 million ratings provided by 480K randomly-chosen, anonymous Netflix customers for 17K movie titles. The rating scale is from 1 to 5. In this experiment we report the root mean square error (RMSE) 5.28 and compare to [445, 423]. The results are shown in Fig. 5.14. Using harmonic analysis outperformed the reported results of Mixed Matrix Factorization [445] and SVD [423] by 3% and 5% respectively. It is important to note that our

⁴<http://www.lifecrunch.biz/archives/207>

method only relies on the provided ratings while the other two methods make use of context data describing the movies and the users.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (p_u - q_i)^2}{N}} \quad (5.28)$$

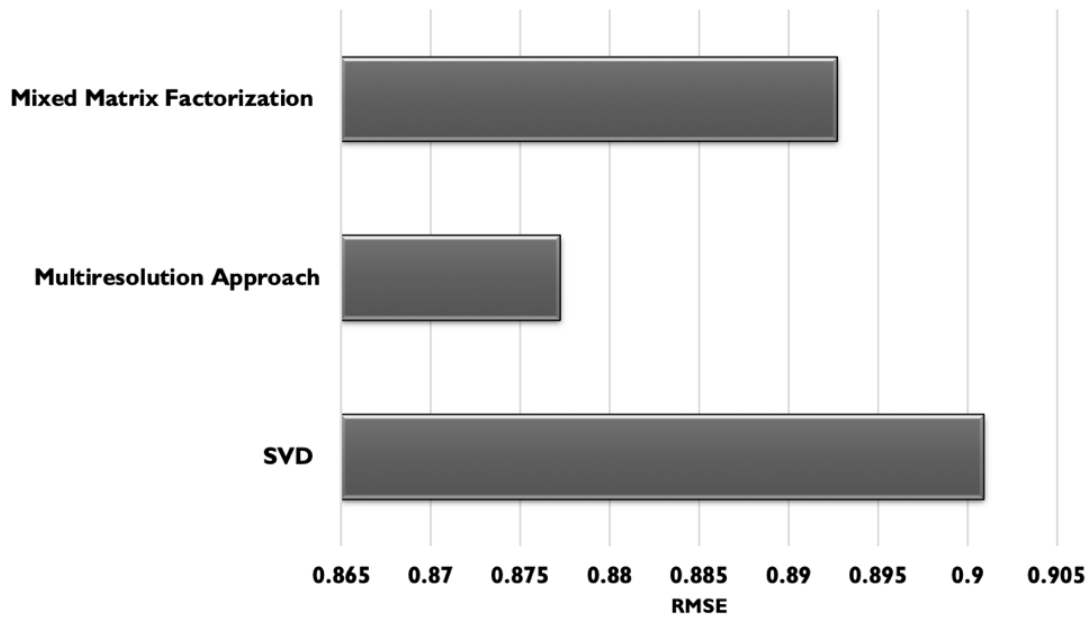


Figure 5.14: RMSE measure for different approaches using Netflix dataset.

5.4.3 Summary

In this section, we proposed a comprehensive coverage of a new multiresolution method for recommender systems based on Harmonic Analysis. The method improves the accuracy of systems with sparse user-item ratings. Experiments with both Movie Lens and Netflix data showed improvements in accuracy of the recommender system compared to conventional user-based and item-based collaborative filtering by 40%. Moreover, the proposed approach outperformed state of the art approaches in terms of accuracy and time per iteration.

Chapter 6

Conclusion and Guidelines for Future Work

Several contributions presented in the dissertation can be further extended to produce better outcomes. We highlight in this section ideas for integrating sentiment mining models into recommender system models. While the presented approaches for recommender systems helped in improving the accuracy and in addressing the sparsity challenge, the issue of cold start users remain unresolved. Cold-start users are users who did not provide any ratings and hence the recommender system cannot provide accurate recommendations. To address this problem, we suggest to use the opinion of users and integrate sentiment classification models into recommender systems. The flow of the proposed future work in alignment with the dissertation contributions is shown in Fig. 6.1.

Based on the assumption that cold start users are most likely active on social network and have provided several textual information charged with sentiment, and given the advances in opinion mining models, one could benefit from this knowledge to address the challenge of cold start users. To address the challenge of cold start users, the system needs additional information such as the context of the user and most importantly the sentiment of the user extracted from the textual information of his/her social media activity. By analyzing his/her sentiment, the system would be able to learn his/her preferences: what he/she likes and what he/she does not like and the system would be able to compare and contrast his/her sentiment to other users without the use of the ratings. By including the sentiment, the model would be able to provide cold start users an improved user recommendation as suggested by [433] for recommending education content and [435] for multimedia recommendations, for instance. Previous work showed that including textual sentiment information results in improved recommendation compared to predictions derived from user ratings only [434, 29].

Although integrating sentiment into recommender systems have been

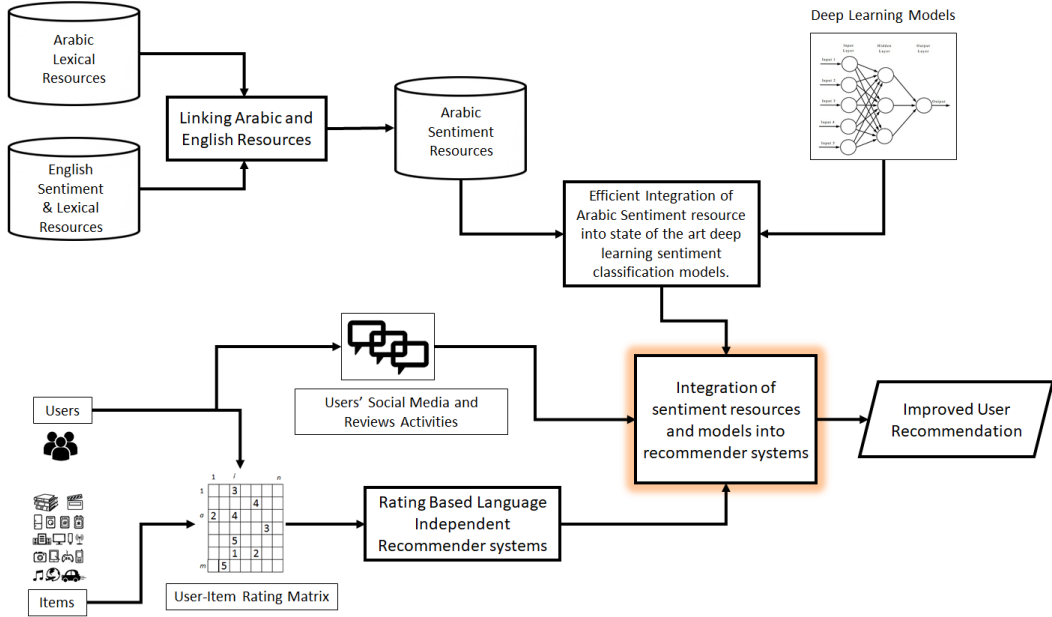


Figure 6.1: Future work highlighted in orange in alignment with achieved contributions.

explored, the work has still a lot of room for improvement. First, the sentiment analysis models used are rather basic with average accuracy. Second, only English textual data is analyzed and there is no work that integrates Arabic text. Thus, a possible future direction is the integration of an improved Arabic sentiment classification model. In order to integrate sentiment into recommender systems, two steps are needed. The first step is to map the output of the sentiment classifier to a rating score s_i provided by a user u to an item i . The second step is to combine the score s_{ui} with the rating score r_{ui} predicted using best approach of context independent recommender systems presented in sections 5.3 and 5.4. Mathematically, one can express the final predicted rating value p_i as in 6.1.

$$p_i = \min(1, r_{ui}) * \gamma * r_{ui} + \min(1, s_{ui}) * \delta * s_{ui} \quad (6.1)$$

The \min function is used to integrate all potential type of users: users who provided ratings and reviews, users who provided ratings only, or cold start users who provided reviews. The goal is to find the weights γ and δ that minimize the following optimization problem (6.2) subject to $1 \leq p_i \leq R$ where q_i is the accurate rating value from training data and R is the maximum rating value of a rating interval. For example, for a rating interval between 1 and 5 R is equal to 5.

$$\underset{\gamma, \delta}{\operatorname{argmin}} \sum_{i=1}^N \frac{|p_i - q_i|}{N} \quad (6.2)$$

In conclusion, we presented in this dissertation several resources and analytics to help in daily-life decisions. The decision system consists of a combination of accurate Arabic opinion mining models and scalable, and improved context and language independent recommender systems. A detailed literature review was conducted about the different components essential for an Arabic opinion mining system including Arabic NLP tools, Arabic lexical resources, opinion mining models. Insights and research roadmap for Arabic opinion mining were also provided. Moreover, details about existing applications and recommender systems' models were also presented. As a result of the work on this dissertation several contributions were achieved: a total 2 transactions and 14 conference papers were published. Multiple large scale lexical resources were developed using heuristic approaches and machine learning approaches. ArSenL [36] was developed using a heuristic approach by linking multiple existing Arabic resources to English sentiment resources. When used in an opinion mining task, ArSenL outperformed existing sentiment resources for subjectivity detection and polarity classification. Several success usage stories of ArSenL were reported with more than 95 citations. For instance, ArSenL was integrated into state-of-the-art deep learning model for sentiment classification, namely RAE [11]. AROMA [31] outperformed other deep learning models for Arabic sentiment classification on different datasets. ArSenL was also used in the design of a light lexicon-based mobile application for sentiment mining of Arabic tweets [37]. The application provides the user with different functionalities and graphical representations to automatically retrieve and classify the sentiment of the latest tweets for a given topic. EmoWordNet [43], a large scale emotion lexicon, was also developed by expanding an existing automatically developed emotion lexicon, Depechemood [130], using synonymy relationship from EWN [7]. EmoWordNet performed better than DepecheMood when utilized for emotion recognition from text. Since both ArSenL and EmoWordNet are linked to EWN synsets, we also presented ArSEL, an Arabic Sentiment and Emotion lexicon, that includes more than 94% of ArSenL lemmas annotated with 8 emotion scores in addition to the 3 sentiment scores. When used in an emotion classification task, ArSEL outperformed a majority baseline with an increase of 69.77% in average F1. In addition to the heuristic approaches used for lexical expansion, machine learning approaches were also evaluated in the objective of expanding AWN. Specifically, link prediction was used to develop ArSenL 2.0. A gold dataset consisting of accurate links between SAMA lemmas and EWN synsets was used to evaluate the performance of several similarity measures for AWN expansion. Jaccard similarity and cosine similarity were used with SAMA gloss terms, machine translation tables, word embeddings, EWN synset terms, glosses and extended glosses. A detailed error analysis was performed for the different link prediction methods. Advantages and limitations of each technique were assessed and a list of reasons resulting in false positives and false negatives was compiled

and supported with examples. Finally, a combination of the best two techniques was proposed for lexical expansion and enrichment with EWN cognitive and semantic lexical relations. ArSenL 2.0 achieved better accuracy results than ArSenL in terms of accurate meaning representation between Arabic lemmas and EWN synsets. ArSenL 2.0 also slightly outperformed ArSenL when used for sentiment and subjectivity classification. As presented in [38, 44], the expertise in opinion mining allowed us to achieve the first position in different subtasks of SemEval 2017 task 4: Sentiment Analysis in Twitter [205] and SemEval 2018 task1: Affect in Tweets [361].

With an overall objective of using opinion mining in real world applications to enhance recommendation, we developed two approaches for improving the accuracy of rating prediction using collaborative filtering. Using a hybrid approach combining user-based and item-based collaborative filtering [32] proved to perform better than typical baseline approaches. In order to address the sparsity challenge of recommender systems, we also proposed a multiresolution approach to compute the missing ratings and improve the recommendation in a scalable approach. We were able to achieve better accuracy by 40% compared to baseline approaches. Moreover, the multiresolution approach outperformed existing state-of-the-art methods in terms of accuracy and time complexity per iteration.

To address the cold start user challenge of recommender systems, future work would need to consider integrating the developed state-of-the-art sentiment models with the proposed context independent recommender systems model. Guidelines were presented in the dissertation in an attempt to formulate the problem as an optimization problem. In terms of lexical resources, there is still room for improving AWN expansion or more generally lexical expansion by incorporating more sophisticated deep learning models for link prediction.

Appendix A

Abbreviations

AI	Artificial Intelligence
AWN	Arabic WordNet
BAMA	Buckwalter Arabic Morphological Analyzer
BLSTM	Bidirection Long Short-Term Memory
BPC	Base Phrase Chunking
CATiB	Columbia Arabic Treebank
CBOW	Continuous Bag Of Words
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DA	Dialectal Arabic
DAE	Deep Auto Encoder
DBN	Deep Belief Network
DNN	Deep Neural Network
ESWN	English SentiWordNet
EWN	English WordNet
GRNN	Gated Recurrent Neural Network
HMM	Hidden Markov Model
LDC	Linguistic Data Consortium
LSTM	Long Short-Term Memory
MADAMIRA	A fast, comprehensive tool for morphological analysis and disambiguation of Arabic
MLP	Multilayer Perceptron
MOOC	Massive Open Online Course
MSA	Modern Standard Arabic
NER	Named Entity Recognition
NLP	Natural Language Processing
OMA	Opinion Mining in Arabic
OOV	Out Of Vocabulary
PATB	Penn Arabic Treebank
POS	Part Of Speech

RAE	Recursive Auto Encoder
RBF	Radial Basis Function
RNTN	Recursive Neural Tensor Network
SAMA	Standard Arabic Morphological Analyzer
SVM	Support Vector Machines
SWN	SentiWordNet

Bibliography

- [1] N. Habash, “Large Scale Lexeme Based Arabic Morphological Generation,” in *JEP-TALN 2004, Session Traitement Automatique de l’Arabe*, (Fes, Morocco), April 2004 2004.
- [2] N. Habash, “Arabic Morphological Representations for Machine Translation,” in *Arabic Computational Morphology: Knowledge-based and Empirical Methods* (A. van den Bosch and A. Soudi, eds.), Kluwer/Springer, 2007.
- [3] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, “The State-of-the-art in Personalized Recommender Systems for Social Networking,” *Artificial Intelligence Review*, vol. 37, no. 2, pp. 119–132, 2012.
- [4] B. Pang, L. Lee, *et al.*, “Opinion Mining and Sentiment Analysis,” *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.
- [5] X. Ding, B. Liu, and P. S. Yu, “A Holistic Lexicon-based Approach to Opinion Mining,” in *Proceedings of the 2008 international conference on web search and data mining*, pp. 231–240, ACM, 2008.
- [6] B. Liu and L. Zhang, “A Survey of Opinion Mining and Sentiment Analysis,” in *Mining text data*, pp. 415–463, Springer, 2012.
- [7] C. Fellbaum, *WordNet*. Wiley Online Library, 1998.
- [8] A. Esuli and F. Sebastiani, “SentiWordNet: a High-coverage Lexical Resource for Opinion Mining,” *Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR)*, 2006.
- [9] S. Baccianella, A. Esuli, and F. Sebastiani, “SentiWordNet 3.0: an Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining,” in *LREC*, vol. 10, pp. 2200–2204, 2010.
- [10] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, *et al.*, “Recursive Deep Models for Semantic Compositionality

- over a Sentiment Treebank,” in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631, p. 1642, 2013.
- [11] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, “Semi-supervised Recursive Autoencoders for Predicting Sentiment Distributions,” in *Proceedings of the conference on empirical methods in natural language processing*, pp. 151–161, Association for Computational Linguistics, 2011.
- [12] I. I. T. Reports, “Internet World Users by Language. Top 10 Languages.” <https://www.internetworldstats.com/stats7.htm>. Accessed: 2019-11-15.
- [13] N. Y. Habash, “Introduction to Arabic Natural Language Processing,” *Synthesis Lectures on Human Language Technologies*, vol. 3, no. 1, pp. 1–187, 2010.
- [14] G. Alwakid, T. Osman, and T. Hughes-Roberts, “Challenges in Sentiment Analysis for Arabic Social Networks,” *Procedia Computer Science*, vol. 117, pp. 89–100, 2017.
- [15] L. Albraheem and H. S. Al-Khalifa, “Exploring the Problems of Sentiment Analysis in Informal Arabic,” in *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*, pp. 415–418, ACM, 2012.
- [16] L. Al-Horaibi and M. B. Khan, “Sentiment Analysis of Arabic Tweets Using Semantic Resources,” *International Journal of Computing & Information Sciences*, vol. 12, no. 2, p. 149, 2016.
- [17] S. Alhazmi, W. Black, and J. McNaught, “Arabic SentiWordNet in Relation to SentiWordNet 3.0,” *2180*, vol. 1266, no. 4, p. 1, 2013.
- [18] M. Abdul-Mageed and M. Diab, “Toward Building a Large-scale Arabic Sentiment Lexicon,” in *Proceedings of the 6th International Global WordNet Conference*, pp. 18–22, 2012.
- [19] M. Elarnaoty, S. AbdelRahman, and A. Fahmy, “A Machine Learning Approach for Opinion Holder Extraction in Arabic Language,” *arXiv preprint arXiv:1206.1011*, 2012.
- [20] W. Black, S. Elkateb, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, and C. Fellbaum, “Introducing the Arabic WordNet Project,” in *Proceedings of the third international WordNet conference*, pp. 295–300, 2006.
- [21] M. Maamouri, D. Graff, B. Bouziri, S. Krouna, A. Bies, and S. Kulick, “Standard Arabic Morphological Analyzer (SAMA) Version 3.1,” *Linguistic Data Consortium, Catalog No.: LDC2010L01*, 2010.

- [22] I. W. Stats, “Usage and Population Statistics.” <https://www.internetworldstats.com/stats1.htm>. Accessed: 2018-12-31.
- [23] L. Martinez, R. M. Rodriguez, and M. Espinilla, “Reja: a Georeferenced Hybrid Recommender System for Restaurants,” in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03*, pp. 187–190, IEEE Computer Society, 2009.
- [24] H. Tan and H. Ye, “A Collaborative Filtering Recommendation Algorithm Based on Item Classification,” in *2009 Pacific-Asia Conference on Circuits, Communications and Systems*, pp. 694–697, May 2009.
- [25] D. A. Al Qudah, A. I. Cristea, S. H. Bazdarevic, S. Al-Saqqa, A. Rodan, and W. Yang, “Personalized E-Advertisement and Experience: Recommending User Targeted Ads,” in *e-Business Engineering (ICEBE), 2015 IEEE 12th International Conference on*, pp. 56–61, IEEE, 2015.
- [26] G. Adomavicius and A. Tuzhilin, “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-art and Possible Extensions,” *IEEE Transactions on Knowledge & Data Engineering*, no. 6, pp. 734–749, 2005.
- [27] X. Su and T. M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” *Advances in artificial intelligence*, vol. 2009, 2009.
- [28] G. Adomavicius and A. Tuzhilin, “Context-aware Recommender Systems,” in *Recommender systems handbook*, pp. 217–253, Springer, 2011.
- [29] G. Ganu, Y. Kakodkar, and A. Marian, “Improving the Quality of Predictions Using Textual Information in Online User Reviews,” *Information Systems*, vol. 38, no. 1, pp. 1–15, 2013.
- [30] G. Badaro, R. Baly, H. Hajj, W. El-Hajj, K. B. Shaban, N. Habash, A. Al-Sallab, and A. Hamdi, “A Survey of Opinion Mining in Arabic: a Comprehensive System Perspective Covering Challenges and Advances in Tools, Resources, Models, Applications, and Visualizations,” *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 18, no. 3, p. 27, 2019.
- [31] A. Al-Sallab, R. Baly, H. Hajj, K. B. Shaban, W. El-Hajj, and G. Badaro, “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)*, vol. 16, no. 4, p. 25, 2017.

- [32] G. Badaro, H. Hajj, W. El-Hajj, and L. Nachman, “A Hybrid Approach with Collaborative Filtering for Recommender Systems,” in *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pp. 349–354, IEEE, 2013.
- [33] G. Badaro, H. Hajj, A. Haddad, W. El-Hajj, and K. B. Shaban, “Recommender Systems Using Harmonic Analysis,” in *2014 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 1004–1011, IEEE, 2014.
- [34] G. Badaro, H. Hajj, A. Haddad, W. El-Hajj, and K. B. Shaban, “A Multiresolution Approach to Recommender Systems,” in *Proceedings of the 8th Workshop on Social Network Mining and Analysis*, p. 9, ACM, 2014.
- [35] G. Badaro, R. Baly, H. Hajj, N. Habash, W. El-hajj, and K. Shaban, “An Efficient Model for Sentiment Classification of Arabic Tweets on Mobiles,” in *Qatar Foundation Annual Research Conference*, no. 1, p. ITPP0631, 2014.
- [36] G. Badaro, R. Baly, H. Hajj, N. Habash, and W. El-Hajj, “A Large Scale Arabic Sentiment Lexicon for Arabic Opinion Mining,” *ANLP 2014*, vol. 165, 2014.
- [37] G. Badaro, R. Baly, R. Akel, L. Fayad, J. Khairallah, H. Hajj, W. El-Hajj, and K. B. Shaban, “A Light Lexicon-based Mobile Application for Sentiment Mining of Arabic Tweets,” in *ANLP Workshop 2015*, p. 18, 2015.
- [38] R. Baly, G. Badaro, A. Hamdi, R. Moukalled, R. Aoun, G. El-Khoury, A. Al Sallab, H. Hajj, N. Habash, K. Shaban, *et al.*, “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). Association for Computational Linguistics, Vancouver, Canada*, pp. 601–608, 2017.
- [39] R. Baly, G. Badaro, G. El-Khoury, R. Moukalled, R. Aoun, H. Hajj, W. El-Hajj, N. Habash, and K. B. Shaban, “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)*, p. 110, 2017.
- [40] L. Constantine, G. Badaro, H. Hajj, W. El-Hajj, L. Nachman, M. BenSaleh, and A. Obeid, “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)*, 2016.

- [41] N. Al-Twairesh, H. Al-Khalifa, and A. Al-Salman, “Towards Analyzing Saudi Tweets,” in *Arabic Computational Linguistics (ACLing), 2015 First International Conference on*, pp. 114–117, IEEE, 2015.
- [42] R. Georges Baly, G. Badaro, H. Hajj, N. Habash, W. El Hajj, and K. Shaban, “Semantic Model Representation for Human’s Pre-conceived Notions In Arabic Text with Applications to Sentiment Mining,” in *Qatar Foundation Annual Research Conference*, no. 1, p. ITPP1075, 2014.
- [43] G. Badaro, H. Jundi, H. Hajj, and W. El-Hajj, “EmoWordNet: Automatic Expansion of Emotion Lexicon Using English WordNet,” in *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pp. 86–93, 2018.
- [44] G. Badaro, O. El Jundi, A. Khaddaj, A. Maarouf, R. Kain, H. Hajj, and W. El-Hajj, “EMA at SemEval-2018 Task 1: Emotion Mining for Arabic,” in *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 236–244, 2018.
- [45] G. Badaro, H. Jundi, H. Hajj, W. El-Hajj, and N. Habash, “ArSEL: a Large Scale Arabic Sentiment and Emotion Lexicon,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (H. Al-Khalifa, K. S. University, K. W. Magdy, U. of Edinburgh, U. K. Darwish, Q. C. R. Institute, Q. T. Elsayed, Q. University, and Qatar, eds.), (Paris, France), European Language Resources Association (ELRA), may 2018.
- [46] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, “The Penn Arabic Treebank: Building a Large-scale Annotated Arabic Corpus,” in *NEMLAR conference on Arabic language resources and tools*, vol. 27, pp. 466–467, 2004.
- [47] N. Habash and R. M. Roth, “CATiB: the Columbia Arabic Treebank,” in *Proceedings of the ACL-IJCNLP 2009 conference short papers*, pp. 221–224, Association for Computational Linguistics, 2009.
- [48] M. Diab, K. Hacioglu, and D. Jurafsky, “Automatic Tagging of Arabic Text: from Raw Text to Base Phrase Chunks,” in *Proceedings of HLT-NAACL 2004: Short papers*, pp. 149–152, Association for Computational Linguistics, 2004.
- [49] N. Habash and O. Rambow, “Arabic Tokenization, Part-of-speech Tagging and Morphological Disambiguation in One Fell Swoop,” in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 573–580, Association for Computational Linguistics, 2005.

- [50] I. Zitouni, J. S. Sorensen, and R. Sarikaya, “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*, pp. 577–584, Association for Computational Linguistics, 2006.
- [51] S. Green and C. D. Manning, “Better Arabic Parsing: Baselines, Evaluations, and Analysis,” in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 394–402, Association for Computational Linguistics, 2010.
- [52] Y. Marton, N. Habash, and O. Rambow, “Dependency Parsing of Modern Standard Arabic with Lexical and Inflectional Features,” *Computational Linguistics*, vol. 39, no. 1, pp. 161–194, 2013.
- [53] Y. Samih, M. Attia, M. Eldesouki, H. Mubarak, A. Abdelali, L. Kallmeyer, and K. Darwish, “A Neural Architecture for Dialectal Arabic Segmentation,” *WANLP 2017 (co-located with EACL 2017)*, p. 46, 2017.
- [54] K. Shaalan and H. Raza, “NERA: Named Entity Recognition for Arabic,” *Journal of the Association for Information Science and Technology*, vol. 60, no. 8, pp. 1652–1663, 2009.
- [55] M. Oudah and K. Shaalan, “NERA 2.0: Improving Coverage and Performance of Rule-based Named Entity Recognition for Arabic,” *Natural Language Engineering*, vol. 23, no. 3, pp. 441–472, 2017.
- [56] H. A. Bakr, K. Shaalan, and I. Ziedan, “A Hybrid Approach for Converting Written Egyptian Colloquial Dialect into Diacritized Arabic,” in *The 6th international conference on informatics and systems, infos2008. cairo university*, 2008.
- [57] A. Pasha, M. Al-Badrashiny, M. T. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth, “MADAMIRA: a Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic,” in *LREC*, vol. 14, pp. 1094–1101, 2014.
- [58] N. Habash, O. Rambow, and R. Roth, “MADA+ TOKAN: a Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization,” in *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, vol. 41, p. 62, 2009.
- [59] M. Diab, “Second Generation AMIRA Tools for Arabic Processing: Fast and Robust Tokenization, POS Tagging, and Base Phrase Chunking,” in

- 2nd International Conference on Arabic Language Resources and Tools*, vol. 110, 2009.
- [60] M. N. Ibrahim, M. N. Mahmoud, and D. A. El-Reedy, “Bel-Arabi: Advanced Arabic Grammar Analyzer,” *International Journal of Social Science and Humanity*, vol. 6, no. 5, p. 341, 2016.
- [61] T. Buckwalter, “Buckwalter Arabic Morphological Analyzer (BAMA) Version 2.0. Linguistic Data Consortium (LDC) Catalogue Number LDC2004L02,” tech. rep., ISBN1-58563-324-0, 2004.
- [62] M. Popel and Z. Zabokrtský, “TectoMT: Modular NLP Framework,” *IceTAL*, vol. 6233, pp. 293–304, 2010.
- [63] M. Abdul-Mageed, M. Diab, and S. Kübler, “Asma: a System for Automatic Segmentation and Morpho-syntactic Disambiguation of Modern Standard Arabic,” in *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pp. 1–8, 2013.
- [64] H. Kilany, H. Gadalla, H. Arram, A. Yacoub, A. El-Habashi, and C. McLemore, “Egyptian Colloquial Arabic Lexicon,” *LDC catalog number LDC99L22*, 2002.
- [65] K. Duh and K. Kirchhoff, “POS Tagging of Dialectal Arabic: a Minimally Supervised Approach,” in *Proceedings of the acl workshop on computational approaches to semitic languages*, pp. 55–62, Association for Computational Linguistics, 2005.
- [66] E. Mohamed, B. Mohit, and K. Oflazer, “Annotating and Learning Morphological Segmentation of Egyptian Colloquial Arabic,” in *LREC*, pp. 873–877, 2012.
- [67] R. Al-Sabbagh and R. Girju, “A Supervised POS Tagger for Written Arabic Social Networking Corpora,” in *KONVENS*, pp. 39–52, 2012.
- [68] N. Habash, R. Eskander, and A. Hawwari, “A Morphological Analyzer for Egyptian Arabic,” in *Proceedings of the twelfth meeting of the special interest group on computational morphology and phonology*, pp. 1–9, Association for Computational Linguistics, 2012.
- [69] H. Elfardy, M. Al-Badrashiny, and M. Diab, “AIDA: Identifying Code Switching in Informal Arabic Text,” *EMNLP 2014*, p. 94, 2014.
- [70] S. Khalifa, S. Hassan, and N. Habash, “A Morphological Analyzer for Gulf Arabic Verbs,” *WANLP 2017 (co-located with EACL 2017)*, p. 35, 2017.

- [71] D. Chiang, M. T. Diab, N. Habash, O. Rambow, and S. Shareef, “Parsing Arabic Dialects,” in *EACL*, 2006.
- [72] W. Salloum and N. Habash, “Dialectal to Standard Arabic Paraphrasing to Improve Arabic-English Statistical Machine Translation,” in *Proceedings of the first workshop on algorithms and resources for modelling of dialects and language varieties*, pp. 10–21, Association for Computational Linguistics, 2011.
- [73] H. Elfardy and M. T. Diab, “Token Level Identification of Linguistic Code Switching,” in *COLING (Posters)*, pp. 287–296, 2012.
- [74] O. F. Zaidan and C. Callison-Burch, “Arabic Dialect Identification,” *Computational Linguistics*, vol. 40, no. 1, pp. 171–202, 2014.
- [75] Y. Belinkov and J. Glass, “A Character-level Convolutional Neural Network for Distinguishing Similar Languages and Dialects,” *arXiv preprint arXiv:1609.07568*, 2016.
- [76] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, “Aravec: a Set of Arabic Word Embedding Models for Use in Arabic NLP,” *Procedia Computer Science*, vol. 117, pp. 256–265, 2017.
- [77] M. Abdul-Mageed, M. T. Diab, and M. Korayem, “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*, pp. 587–591, Association for Computational Linguistics, 2011.
- [78] M. Abdul-Mageed and M. Korayem, “Automatic Identification of Subjectivity in Morphologically Rich Languages: the Case of Arabic,” *Computational approaches to subjectivity and sentiment analysis*, vol. 2, 2010.
- [79] M. Abdul-Mageed and M. T. Diab, “Subjectivity and Sentiment Annotation of Modern Standard Arabic Newswire,” in *Proceedings of the 5th linguistic annotation workshop*, pp. 110–118, Association for Computational Linguistics, 2011.
- [80] R. Feldman, “Techniques and Applications for Sentiment Analysis,” *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.
- [81] M. Al-Kabi, A. Gigieh, I. Alsmadi, H. Wahsheh, and M. Haidar, “An Opinion Analysis Tool for Colloquial and Standard Arabic,” in *The Fourth International Conference on Information and Communication Systems (ICICS 2013)*, pp. 23–25, 2013.

- [82] K. Al-Rowaily, M. Abulaish, N. A.-H. Haldar, and M. Al-Rubaian, “BiSAL—A Bilingual Sentiment Analysis Lexicon to Analyze Dark Web Forums for Cyber Security,” *Digital Investigation*, vol. 14, pp. 53–62, 2015.
- [83] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub, “Arabic Sentiment Analysis: Lexicon-based and Corpus-based,” in *Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on*, pp. 1–6, IEEE, 2013.
- [84] M. Thelwall, “Heart and Soul: Sentiment Strength Detection in the Social Web with SentiStrength,” *Proceedings of the CyberEmotions*, vol. 5, pp. 1–14, 2013.
- [85] S. R. El-Beltagy, “NileULex: a Phrase and Word Level Sentiment Lexicon for Egyptian and Modern Standard Arabic,” in *LREC*, 2016.
- [86] S. R. El-Beltagy and A. Ali, “Open Issues in the Sentiment Analysis of Arabic Social Media: a Case Study,” in *Innovations in information technology (iit), 2013 9th international conference on*, pp. 215–220, IEEE, 2013.
- [87] S. R. El-Beltagy, “Niletmrg at SemEval-2016 Task 7: Deriving Prior Polarities for Arabic Sentiment Terms,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 486–490, 2016.
- [88] S. Kiritchenko, S. Mohammad, and M. Salameh, “Semeval-2016 Task 7: Determining Sentiment Intensity of English and Arabic Phrases,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 42–51, 2016.
- [89] S. R. El-Beltagy, “WeightedNileULex: A Scored Arabic Sentiment Lexicon for Improved Sentiment Analysis,” *Language Processing, Pattern Recognition and Intelligent Systems. Special Issue on Computational Linguistics, Speech& Image Processing for Arabic Language. World Scientific Publishing Co*, 2017.
- [90] H. S. Ibrahim, S. M. Abdou, and M. Gheith, “Idioms-proverbs Lexicon for Modern Standard Arabic and Colloquial Sentiment Analysis,” *arXiv preprint arXiv:1506.01906*, 2015.
- [91] A. Mourad and K. Darwish, “Subjectivity and Sentiment Analysis of Modern Standard Arabic and Arabic Microblogs,” in *WASSA@ NAACL-HLT*, pp. 55–64, 2013.

- [92] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing Contextual Polarity in Phrase-level Sentiment Analysis,” in *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pp. 347–354, Association for Computational Linguistics, 2005.
- [93] A. Hassan, A. Abu-Jbara, R. Jha, and D. Radev, “Identifying the Semantic Orientation of Foreign Words,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pp. 592–597, Association for Computational Linguistics, 2011.
- [94] F. H. Mahyoub, M. A. Siddiqui, and M. Y. Dahab, “Building an Arabic Sentiment Lexicon Using Semi-supervised Learning,” *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 4, pp. 417–424, 2014.
- [95] P. D. Turney and M. L. Littman, “Unsupervised Learning of Semantic Orientation from a Hundred-billion-word Corpus,” *arXiv preprint cs/0212012*, 2002.
- [96] R. Eskander and O. Rambow, “SLSA: a Sentiment Lexicon for Standard Arabic,” in *EMNLP*, pp. 2545–2550, 2015.
- [97] T. Buckwalter, “Arabic Morphological Analyzer (AraMorph),” 2002.
- [98] K. S. Sabra, R. N. Zantout, M. A. El Abed, and L. Hamandi, “Sentiment Analysis: Arabic Sentiment Lexicons,” in *Sensors Networks Smart and Emerging Technologies (SENSET), 2017*, pp. 1–4, 2017.
- [99] N. Abdulla, S. Mohammed, M. Al-Ayyoub, M. Al-Kabi, *et al.*, “Automatic Lexicon Construction for Arabic Sentiment Analysis,” in *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, pp. 547–552, IEEE, 2014.
- [100] M. Abdul-Mageed and M. T. Diab, “SANA: a Large Scale Multi-Genre, Multi-Dialect Lexicon for Arabic Subjectivity and Sentiment Analysis,” in *LREC*, pp. 1162–1169, 2014.
- [101] M. T. Diab, M. Al-Badrashiny, M. Aminian, M. Attia, H. Elfardy, N. Habash, A. Hawwari, W. Salloum, P. Dasigi, and R. Eskander, “Tharwa: a Large Scale Dialectal Arabic-Standard Arabic-English Lexicon,” in *LREC*, pp. 3782–3789, 2014.
- [102] D. R. Heise, *Expressive Order: Confirming Sentiments in Social Actions*. Springer Science & Business Media, 2007.

- [103] M. Abdul-Mageed, M. Korayem, and A. YoussefAgha, “Yes We Can?: Subjectivity Annotation and Tagging for the Health Domain,” 2011.
- [104] Y. Chen and S. Skiena, “Building sentiment Lexicons for All Major Languages,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 383–389, 2014.
- [105] M. Al-Ayyoub, S. B. Essa, and I. Alsmadi, “Lexicon-based Sentiment Analysis of Arabic Tweets,” *International Journal of Social Network Mining*, vol. 2, no. 2, pp. 101–114, 2015.
- [106] “Dataset for Arabic Document Classification.” <http://diab.edublogs.org/dataset-for-arabic-document-classification/>. Accessed: 2017-08-22.
- [107] H. S. Ibrahim, S. M. Abdou, and M. Gheith, “Sentiment Analysis for Modern Standard Arabic and Colloquial,” *arXiv preprint arXiv:1505.03105*, 2015.
- [108] H. S. Ibrahim, S. M. Abdou, and M. Gheith, “Automatic Expandable Large-scale Sentiment Lexicon of Modern Standard Arabic and Colloquial,” in *Arabic Computational Linguistics (ACLing), 2015 First International Conference on*, pp. 94–99, IEEE, 2015.
- [109] H. ElSahar and S. R. El-Beltagy, “A Fully Automated Approach for Arabic Slang Lexicon Extraction from Microblogs,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 79–91, Springer, 2014.
- [110] H. ElSahar and S. R. El-Beltagy, “Building Large Arabic Multi-domain Resources for Sentiment Analysis,” in *CICLing (2)*, pp. 23–34, 2015.
- [111] T. Al-Moslmi, M. Albared, A. Al-Shabi, N. Omar, and S. Abdullah, “Arabic Senti-lexicon: Constructing Publicly Available Language Resources for Arabic Sentiment Analysis,” *Journal of Information Science*, p. 0165551516683908, 2017.
- [112] N. Al-Twairesh, H. Al-Khalifa, and A. AlSalman, “AraSenTi: Large-scale Twitter-specific Arabic Sentiment Lexicons,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 697–705, 2016.
- [113] M. Hu and B. Liu, “Mining and Summarizing Customer Reviews,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, ACM, 2004.

- [114] N. Al-Twairesh, H. Al-Khalifa, A. Al-Salman, and Y. Al-Ohali, “AraSenTi-Tweet: a Corpus for Arabic Sentiment Analysis of Saudi Tweets,” *Procedia Computer Science*, vol. 117, pp. 63–72, 2017.
- [115] M. Nabil, M. A. Aly, and A. F. Atiya, “ASTD: Arabic Sentiment Tweets Dataset,” in *EMNLP*, pp. 2515–2519, 2015.
- [116] E. Refaee and V. Rieser, “An Arabic Twitter Corpus for Subjectivity and Sentiment Analysis,” in *LREC*, pp. 2268–2273, 2014.
- [117] S. Kiritchenko and S. M. Mohammad, “Capturing Reliable Fine-grained Sentiment Associations by Crowdsourcing and Best-worst Scaling,” *arXiv preprint arXiv:1712.01741*, 2017.
- [118] A. Assiri, A. Emam, and H. Al-Dossari, “Towards Enhancement of a Lexicon-based Approach for Saudi Dialect Sentiment Analysis,” *Journal of Information Science*, p. 0165551516688143, 2017.
- [119] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, “Moses: Open Source Toolkit for Statistical Machine Translation,” in *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pp. 177–180, Association for Computational Linguistics, 2007.
- [120] P. Ekman, “An Argument for Basic Emotions,” *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.
- [121] R. Plutchik, *The Psychology and Biology of Emotion*. HarperCollins College Publishers, 1994.
- [122] C. Strapparava, A. Valitutti, *et al.*, “WordNet Affect: an Affective Extension of WordNet,” in *LREC*, vol. 4, pp. 1083–1086, 2004.
- [123] S. M. Mohammad and P. D. Turney, “Crowdsourcing a Word–emotion Association Lexicon,” *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013.
- [124] J. R. L.-B. Bernard and J. R. L.-B. Bernard, *The Macquarie Thesaurus*. Macquarie, 1986.
- [125] P. J. Stone, D. C. Dunphy, and M. S. Smith, “The General Inquirer: a Computer Approach to Content Analysis,” 1966.
- [126] S. M. Mohammad, S. Kiritchenko, and X. Zhu, “NRC-Canada: Building the State-of-the-art in Sentiment Analysis of Tweets,” *arXiv preprint arXiv:1308.6242*, 2013.

- [127] E. Cambria, C. Havasi, and A. Hussain, “SenticNet 2: a Semantic and Affective Resource for Opinion Mining and Sentiment Analysis,” in *FLAIRS conference*, pp. 202–207, 2012.
- [128] H. Liu and P. Singh, “ConceptNet: a Practical Commonsense Reasoning Tool-kit,” *BT technology journal*, vol. 22, no. 4, pp. 211–226, 2004.
- [129] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, “Textual Affect Sensing for Sociable and Expressive Online Communication,” *Affective Computing and Intelligent Interaction*, pp. 218–229, 2007.
- [130] J. Staiano and M. Guerini, “DepecheMood: a Lexicon for Emotion Analysis from Crowd-annotated News,” *arXiv preprint arXiv:1405.1605*, 2014.
- [131] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, “SentiFul: a Lexicon for Sentiment Analysis,” *IEEE Transactions on Affective Computing*, vol. 2, no. 1, pp. 22–36, 2011.
- [132] C. Strapparava and R. Mihalcea, “SemEval-2007 Task 14: Affective Text,” in *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 70–74, Association for Computational Linguistics, 2007.
- [133] A. Bandhakavi, N. Wiratunga, P. Deepak, and S. Massie, “Generating a Word-Emotion Lexicon from# Emotional Tweets,” in ** SEM@ COLING*, pp. 12–21, 2014.
- [134] A. Bandhakavi, N. Wiratunga, S. Massie, and D. Padmanabhan, “Lexicon Generation for Emotion Detection from Text,” *IEEE intelligent systems*, vol. 32, no. 1, pp. 102–108, 2017.
- [135] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, “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)*, pp. 587–592, IEEE, 2012.
- [136] C. Yang, K. H.-Y. Lin, and H.-H. Chen, “Building Emotion Lexicon from Weblog Corpora,” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pp. 133–136, Association for Computational Linguistics, 2007.
- [137] G. Xu, X. Meng, and H. Wang, “Build Chinese Emotion Lexicons Using a Graph-based Algorithm and Multiple Resources,” in *Proceedings of the 23rd international conference on computational linguistics*, pp. 1209–1217, Association for Computational Linguistics, 2010.

- [138] A. Abdaoui, J. Azé, S. Bringay, and P. Poncelet, “Feel: a French Expanded Emotion Lexicon,” *Language Resources and Evaluation*, vol. 51, no. 3, pp. 833–855, 2017.
- [139] D. Shen and M. Lapata, “Using Semantic Roles to Improve Question Answering,” in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.
- [140] J. Prager, J. Chu-Carroll, and K. Czuba, “Use of WordNet Hypernyms for Answering What-Is Questions—DRAFT,” 2001.
- [141] D. I. Moldovan and R. Mihalcea, “Using Wordnet and Lexical Operators to Improve Internet Searches,” *IEEE Internet Computing*, vol. 4, no. 1, pp. 34–43, 2000.
- [142] S. Banerjee and T. Pedersen, “An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet,” in *International conference on intelligent text processing and computational linguistics*, pp. 136–145, Springer, 2002.
- [143] S. Patwardhan and T. Pedersen, “Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts,” in *Proceedings of the Workshop on Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*, 2006.
- [144] A. Esuli and F. Sebastiani, “SentiWordNet: a High-coverage Lexical Resource for Opinion Mining,” *Evaluation*, pp. 1–26, 2007.
- [145] H. Rodríguez, D. Farwell, J. Farreres, M. Bertran, M. Alkhalifa, M. A. Martí, W. Black, S. Elkateb, J. Kirk, A. Pease, *et al.*, “Arabic WordNet: Current State and Future Extensions,” in *Proceedings of The Fourth Global WordNet Conference, Szeged, Hungary*, no. 387-405, 2008.
- [146] H. Rodríguez, D. Farwell, J. Ferreres, M. Bertran, M. Alkhalifa, and M. A. Martí, “Arabic WordNet: Semi-automatic Extensions Using Bayesian Inference,” in *LREC*, 2008.
- [147] M. Alkhalifa and H. Rodríguez, “Automatically Extending NE Coverage of Arabic WordNet Using Wikipedia,” in *Proc. Of the 3rd International Conference on Arabic Language Processing CITALA2009, Rabat, Morocco*, 2009.
- [148] M. Alkhalifa and H. Rodríguez, “Automatically Extending Named Entities Coverage of Arabic WordNet Using Wikipedia,” *International Journal on Information and Communication Technologies*, vol. 3, no. 3, pp. 20–36, 2010.

- [149] L. Abouenour, K. Bouzoubaa, and P. Rosso, “On the Evaluation and Improvement of Arabic WordNet Coverage and Usability,” *Language resources and evaluation*, vol. 47, no. 3, pp. 891–917, 2013.
- [150] M. Aminian, M. Al-Badrashiny, and M. Diab, “Automatic Verification and Augmentation of Multilingual Lexicons,” in *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pp. 73–81, 2016.
- [151] R. Navigli and S. P. Ponzetto, “BabelNet: Building a Very Large Multilingual Semantic Network,” in *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 216–225, Association for Computational Linguistics, 2010.
- [152] R. Navigli and S. P. Ponzetto, “BabelNet: the Automatic Construction, Evaluation and Application of a Wide-coverage Multilingual Semantic Network,” *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.
- [153] W. Guo and M. Diab, “Modeling Sentences in the Latent Space,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 864–872, Association for Computational Linguistics, 2012.
- [154] W. Guo and M. Diab, “Learning the Latent Semantics of a Concept from its Definition,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 140–144, 2012.
- [155] W. Guo and M. Diab, “Improving Lexical Semantics for Sentential Semantics: Modeling Selectional Preference and Similar Words in a Latent Variable Model,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 739–745, 2013.
- [156] A. Abu-Jbara, M. Diab, P. Dasigi, and D. Radev, “Subgroup Detection in Ideological Discussions,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 399–409, Association for Computational Linguistics, 2012.
- [157] P. Dasigi, W. Guo, and M. Diab, “Genre Independent Subgroup Detection in Online Discussion Threads: a Pilot Study of Implicit Attitude Using Latent Textual Semantics,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pp. 65–69, Association for Computational Linguistics, 2012.

- [158] F. Bond and R. Foster, “Linking and Extending an Open Multilingual WordNet,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1352–1362, 2013.
- [159] V. Hanoka and B. Sagot, “WordNet Creation and Extension Made Simple: a Multilingual Lexicon-based Approach Using Wiki Resources,” in *LREC 2012: 8th international conference on Language Resources and Evaluation*, p. 6, 2012.
- [160] B. Sagot and D. Fišer, “Extending Wordnets by Learning from Multiple Resources,” in *LTC’11: 5th Language and Technology Conference*, 2011.
- [161] S. Joshi, A. Chatterjee, A. K. Karra, and P. Bhattacharyya, “Eating your Own Cooking: Automatically Linking WordNet Synsets of Two Languages,” *Proceedings of COLING 2012: Demonstration Papers*, pp. 239–246, 2012.
- [162] K. Patel, D. Kanojia, and P. Bhattacharyya, “Semi-automatic WordNet Linking Using Word Embeddings,” 2018.
- [163] M. Abdul-Mageed and M. T. Diab, “AWATIF: A Multi-Genre Corpus for Modern Standard Arabic Subjectivity and Sentiment Analysis,” in *LREC*, pp. 3907–3914, 2012.
- [164] M. Rushdi-Saleh, M. T. Martín-Valdivia, L. A. Ureña-López, and J. M. Perea-Ortega, “OCA: Opinion Corpus for Arabic,” *Journal of the Association for Information Science and Technology*, vol. 62, no. 10, pp. 2045–2054, 2011.
- [165] M. A. Aly and A. F. Atiya, “LABR: a Large Scale Arabic Book Reviews Dataset,” in *ACL (2)*, pp. 494–498, 2013.
- [166] M. Nabil, M. Aly, and A. Atiya, “LABR: a Large Scale Arabic Sentiment Analysis Benchmark,” *arXiv preprint arXiv:1411.6718*, 2014.
- [167] M. Saad, D. Langlois, and K. Smaïli, “Building and Modelling Multilingual Subjective Corpora,” in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, European Language Resources Association (ELRA), 2014.
- [168] M. Saad, D. Langlois, and K. Smaïli, “Comparing Multilingual Comparable Articles Based on Opinions,” in *Proceedings of the 6th Workshop on Building and Using Comparable Corpora*, pp. 105–111, 2013.

- [169] A. Elnagar and O. Einea, “Brad 1.0: Book Reviews in Arabic Dataset,” in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–8, IEEE, 2016.
- [170] A. Elnagar, Y. S. Khalifa, and A. Einea, “Hotel Arabic-reviews Dataset Construction for Sentiment Analysis Applications,” in *Intelligent Natural Language Processing: Trends and Applications*, pp. 35–52, Springer, 2018.
- [171] N. Farra, K. McKeown, and N. Habash, “Annotating Targets of Opinions in Arabic Using Crowdsourcing,” in *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pp. 89–98, 2015.
- [172] B. Mohit, A. Rozovskaya, N. Habash, W. Zaghoulani, and O. Obeid, “The First QALB Shared Task on Automatic Text Correction for Arabic,” in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pp. 39–47, 2014.
- [173] W. Zaghoulani, B. Mohit, N. Habash, O. Obeid, N. Tomeh, A. Rozovskaya, N. Farra, S. Alkuhlani, and K. Ofazer, “Large Scale Arabic Error Annotation: Guidelines and Framework,” in *LREC*, pp. 2362–2369, 2014.
- [174] M. Al-Smadi, O. Qawasmeh, B. Talafha, and M. Quwaider, “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*, pp. 726–730, IEEE, 2015.
- [175] H. Al-Sarhan, M. Al-So’ud, M. Al-Smadi, M. Al-Ayyoub, and Y. Jararweh, “Framework for Affective News Analysis of Arabic News: 2014 Gaza Attacks Case Study,” in *Information and Communication Systems (ICICS), 2016 7th International Conference on*, pp. 327–332, IEEE, 2016.
- [176] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, “SemEval-2014 task 4: Aspect Based Sentiment Analysis,” in *Proceedings of the 8th international workshop on semantic evaluation (SemEval-2014)*, 2014.
- [177] M. Al-Kabi, M. Al-Ayyoub, I. Alsmadi, and H. Wahsheh, “A Prototype for a Standard Arabic Sentiment Analysis Corpus,” *Int. Arab J. Inf. Technol.*, vol. 13, no. 1A, pp. 163–170, 2016.
- [178] M. N. Al-Kabi, A. A. Al-Qwaqenah, A. H. Gigieh, K. Alsmearat, M. Al-Ayyoub, and I. M. Alsmadi, “Building a Standard Dataset for Arabic Sentiment Analysis: Identifying Potential Annotation Pitfalls,” in *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*, pp. 1–6, IEEE, 2016.

- [179] M. A. Siddiqui, M. Y. Dahab, and O. A. Batarfi, “Building a Sentiment Analysis Corpus With Multifaceted Hierarchical Annotation,” *International Journal of Computational Linguistics (IJCL)*, 2015.
- [180] A. Assiri, A. Emam, and H. Al-Dossari, “Saudi Twitter Corpus for Sentiment Analysis,” *International Journal of Computer, Electrical, Automation, Control and Information Engineering. World Academy of Science, Engineering and Technology*, vol. 10, pp. 242–245, 2016.
- [181] R. M. Duwairi, M. Alfaqeh, M. Wardat, and A. Alrabadi, “Sentiment Analysis for Arabizi Text,” in *Information and Communication Systems (ICICS), 2016 7th International Conference on*, pp. 127–132, IEEE, 2016.
- [182] I. Guellil, A. Adeel, F. Azouaou, and A. Hussain, “Sentialg: Automated Corpus Annotation for Algerian Sentiment Analysis,” in *International Conference on Brain Inspired Cognitive Systems*, pp. 557–567, Springer, 2018.
- [183] S. Medhaffar, F. Bougares, Y. Esteve, and L. Hadrich-Belguith, “Sentiment Analysis of Tunisian Dialects: Linguistic Ressources and Experiments,” in *Proceedings of the Third Arabic Natural Language Processing Workshop*, pp. 55–61, 2017.
- [184] H. S. Ibrahim, S. M. Abdou, and M. Gheith, “MIKA: a Tagged Corpus for Modern Standard Arabic and Colloquial Sentiment Analysis,” in *Recent Trends in Information Systems (ReTIS), 2015 IEEE 2nd International Conference on*, pp. 353–358, IEEE, 2015.
- [185] A. A. Elmadany and W. M. Hamdy Mubarak, “ArSAS: an Arabic Speech-act and Sentiment Corpus of Tweets,” in *OSACT 3: The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools*, p. 20, 2018.
- [186] R. Baly, A. Khaddaj, H. Hajj, W. El-Hajj, and K. Shaban, “ArSentD-LEV: a Multi-Topic Corpus for Target-based Sentiment Analysis in Arabic Levantine Tweets,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (H. Al-Khalifa, K. S. University, K. W. Magdy, U. of Edinburgh, U. K. Darwish, Q. C. R. Institute, Q. T. Elsayed, Q. University, and Qatar, eds.), (Paris, France), European Language Resources Association (ELRA), may 2018.
- [187] A. Hamdi, K. Shaban, and A. Zainal, “Clasenti: a Class-specific Sentiment Analysis Framework,” *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 17, no. 4, p. 32, 2018.

- [188] M. Itani, C. Roast, and S. Al-Khayatt, “Corpora for Sentiment Analysis of Arabic Text in Social Media,” in *Information and Communication Systems (ICICS), 2017 8th International Conference on*, pp. 64–69, IEEE, 2017.
- [189] M. Itani, C. Roast, and S. Al-Khayatt, “Developing Resources for Sentiment Analysis of Informal Arabic Text in Social Media,” *Procedia Computer Science*, vol. 117, pp. 129–136, 2017.
- [190] A. J. S. Al Mukhaiti, S. Siddiqui, and K. Shaalan, “Dataset Built for Arabic Sentiment Analysis,” in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 406–416, Springer, 2017.
- [191] A. Elnagar, O. Einea, and L. Lulu, “Comparative Study of Sentiment Classification for Automated Translated Latin Reviews into Arabic,” in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pp. 443–448, IEEE, 2017.
- [192] N. Ahmed, “Large-scale Arabic Sentiment Corpus and Lexicon Building for Concept-based Sentiment Analysis Systems,” 2018.
- [193] E. Cambria, S. Poria, R. Bajpai, and B. Schuller, “SenticNet 4: a Semantic Resource for Sentiment Analysis Based on Conceptual Primitives,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2666–2677, 2016.
- [194] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, “Ask Me Anything: Dynamic Memory Networks for Natural Language Processing,” in *International Conference on Machine Learning*, pp. 1378–1387, 2016.
- [195] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” *arXiv preprint arXiv:1408.5882*, 2014.
- [196] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. De Luca, and M. Jaggi, “SwissCheese at SemEval-2016 Task 4: Sentiment Classification Using an Ensemble of Convolutional Neural Networks with Distant Supervision,” in *SemEval@ NAACL-HLT*, pp. 1124–1128, 2016.
- [197] F. Lazhar and T. G. Yamina, “Identification of Opinions in Arabic Newspapers,” in *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, pp. 317–319, IEEE, 2010.
- [198] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent Trends in Deep Learning Based Natural Language Processing,” *arXiv preprint arXiv:1708.02709*, 2017.

- [199] A. Shoukry and A. Rafea, “A Hybrid Approach for Sentiment Classification of Egyptian Dialect Tweets,” in *Arabic Computational Linguistics (ACLing), 2015 First International Conference on*, pp. 78–85, IEEE, 2015.
- [200] W. Cherif, A. Madani, and M. Kissi, “A Hybrid Optimal Weighting Scheme and Machine Learning for Rendering Sentiments in Tweets,” *International Journal of Intelligent Engineering Informatics*, vol. 4, no. 3-4, pp. 322–339, 2016.
- [201] N. Al-Twairesh, H. Al-Khalifa, A. Als Salman, and Y. Al-Ohali, “Sentiment Analysis of Arabic Tweets: Feature Engineering and a Hybrid Approach,” *arXiv preprint arXiv:1805.08533*, 2018.
- [202] K. Khalifa and N. Omar, “A Hybrid Method Using Lexicon-based Approach and Naive Bayes Classifier for Arabic Opinion Question Answering,” *Journal of Computer Science*, vol. 10, no. 10, pp. 1961–1968, 2014.
- [203] A. El-Halees, “Arabic Opinion Mining Using Combined Classification Approach,” 2011.
- [204] E. Refaee and V. Rieser, “iLab-Edinburgh at SemEval-2016 Task 7: a Hybrid Approach for Determining Sentiment Intensity of Arabic Twitter Phrases,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 474–480, 2016.
- [205] S. Rosenthal, N. Farra, and P. Nakov, “SemEval-2017 Task 4: Sentiment Analysis in Twitter,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 502–518, 2017.
- [206] H. Awwad and A. Alpkocak, “Performance Comparison of Different Lexicons for Sentiment Analysis in Arabic,” in *2016 Third European Network Intelligence Conference (ENIC)*, pp. 127–133, IEEE, 2016.
- [207] K. Ahmad, D. Cheng, and Y. Almas, “Multi-lingual Sentiment Analysis of Financial News Streams,” in *1st International Workshop on Grid Technology for Financial Modeling and Simulation*, vol. 26, SISSA Medialab, 2007.
- [208] S. Mohammad, M. Salameh, and S. Kiritchenko, “Sentiment Lexicons for Arabic Social Media,” in *LREC*, 2016.
- [209] R. Zbib, E. Malchiodi, J. Devlin, D. Stallard, S. Matsoukas, R. Schwartz, J. Makhoul, O. F. Zaidan, and C. Callison-Burch, “Machine Translation of Arabic Dialects,” in *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pp. 49–59, Association for Computational Linguistics, 2012.

- [210] M. Elhawary and M. Elfeky, “Mining Arabic Business Reviews,” in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pp. 1108–1113, IEEE, 2010.
- [211] A. A. Al-Subaihin, H. S. Al-Khalifa, and A. S. Al-Salman, “A Proposed Sentiment Analysis Tool for Modern Arabic Using Human-based Computing,” in *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, pp. 543–546, ACM, 2011.
- [212] A. S. Al-Subaihin and H. S. Al-Khalifa, “A System for Sentiment Analysis of Colloquial Arabic Using Human Computation,” *The Scientific World Journal*, vol. 2014, 2014.
- [213] S. Siddiqui, A. A. Monem, and K. Shaalan, “Evaluation and Enrichment of Arabic Sentiment Analysis,” in *Intelligent Natural Language Processing: Trends and Applications*, pp. 17–34, Springer, 2018.
- [214] S. Siddiqui, A. A. Monem, and K. Shaalan, “Towards improving sentiment analysis in arabic,” in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016* (A. E. Hassanien, K. Shaalan, T. Gaber, A. T. Azar, and M. F. Tolba, eds.), (Cham), pp. 114–123, Springer International Publishing, 2017.
- [215] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, M. Al-Ayyoub, M. N. Al-Kabi, and S. Al-rifai, “Towards Improving the Lexicon-based Approach for Arabic Sentiment Analysis,” *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 9, no. 3, pp. 55–71, 2014.
- [216] R. Duwairi, N. A. Ahmed, and S. Y. Al-Rifai, “Detecting Sentiment Embedded in Arabic Social Media—a Lexicon-based Approach,” *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 1, pp. 107–117, 2015.
- [217] R. M. Duwairi and M. A. Alshboul, “Negation-aware Framework for Sentiment Analysis in Arabic Reviews,” in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, pp. 731–735, IEEE, 2015.
- [218] S. Oraby, Y. El-Sonbaty, and M. A. El-Nasr, “Finding Opinion Strength Using Rule-based Parsing for Arabic Sentiment Analysis,” in *Mexican International Conference on Artificial Intelligence*, pp. 509–520, Springer, 2013.
- [219] A. Bayoudhi, H. Ghorbel, H. Koubaa, and L. H. Belguith, “Sentiment Classification at Discourse Segment Level: Experiments on Multi-domain Arabic Corpus,” *JLCL*, vol. 30, no. 1, pp. 1–24, 2015.

- [220] I. Obaidat, R. Mohawesh, M. Al-Ayyoub, A.-S. Mohammad, and Y. Jararweh, “Enhancing the Determination of Aspect Categories and their Polarities in Arabic Reviews using Lexicon-based Approaches,” in *Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on*, pp. 1–6, IEEE, 2015.
- [221] M. Al Smadi, I. Obaidat, M. Al-Ayyoub, R. Mohawesh, and Y. Jararweh, “Using Enhanced Lexicon-based Approaches for the Determination of Aspect Categories and their Polarities in Arabic Reviews,” *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 11, no. 3, pp. 15–31, 2016.
- [222] M. Mataoui, O. Zelmati, and M. Boumechache, “A Proposed Lexicon-based Sentiment Analysis Approach for the Vernacular Algerian Arabic,” *Research in Computing Science*, vol. 110, pp. 55–70, 2016.
- [223] H. Mulki, H. Haddad, M. Gridach, and I. Babaoğlu, “Tw-StAR at SemEval-2017 Task 4: Sentiment Classification of Arabic Tweets,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 664–669, 2017.
- [224] M. Salameh, S. Mohammad, and S. Kiritchenko, “Sentiment After Translation: a Case-study on Arabic Social Media Posts,” in *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 767–777, 2015.
- [225] S. M. Mohammad, M. Salameh, and S. Kiritchenko, “How Translation Alters Sentiment,” *J. Artif. Intell. Res. (JAIR)*, vol. 55, pp. 95–130, 2016.
- [226] H. Al Suwaidi, T. R. Soomro, and K. Shaalan, “Sentiment Analysis for Emirati Dialects in Twitter,” *Sindh University Research Journal-SURJ (Science Series)*, vol. 48, no. 4, 2016.
- [227] A. Htait, S. Fournier, and P. Bellot, “LSIS at SemEval-2017 Task 4: Using Adapted Sentiment Similarity Seed Words For English and Arabic Tweet Polarity Classification,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 718–722, 2017.
- [228] T. Khalil, A. Halaby, M. Hammad, and S. R. El-Beltagy, “Which Configuration Works Best? an Experimental Study on Supervised Arabic Twitter Sentiment Analysis,” in *Arabic Computational Linguistics (ACLing), 2015 First International Conference on*, pp. 86–93, IEEE, 2015.

- [229] R. M. Duwairi and I. Qarqaz, “A Framework for Arabic Sentiment Analysis Using Supervised Classification,” *International Journal of Data Mining, Modelling and Management*, vol. 8, no. 4, pp. 369–381, 2016.
- [230] M. N. Al-Kabi, N. A. Abdulla, and M. Al-Ayyoub, “An Analytical Study of Arabic Sentiments: Maktoob Case Study,” in *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*, pp. 89–94, IEEE, 2013.
- [231] A. Shoukry and A. Rafea, “Sentence-level Arabic Sentiment Analysis,” in *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pp. 546–550, IEEE, 2012.
- [232] A. M. Shoukry, “Arabic Sentence-level Sentiment Analysis,” 2013.
- [233] A. Mountassir, H. Benbrahim, and I. Berrada, “A Cross-study of Sentiment Classification on Arabic Corpora,” in *Research and Development in Intelligent Systems XXIX*, pp. 259–272, Springer, 2012.
- [234] N. A. Abdulla, M. Al-Ayyoub, and M. N. Al-Kabi, “An Extended Analytical Study of Arabic Sentiments,” *International Journal of Big Data Intelligence 1*, vol. 1, no. 1-2, pp. 103–113, 2014.
- [235] R. M. Elawady, S. Barakat, and N. M. Elrashidy, “Different Feature Selection for Sentiment Classification,” *International Journal of Information Science and Intelligent System*, vol. 3, no. 1, pp. 137–150, 2014.
- [236] B. Al Shboul, M. Al-Ayyoub, and Y. Jararweh, “Multi-way Sentiment Classification of Arabic Reviews,” in *Information and Communication Systems (ICICS), 2015 6th International Conference on*, pp. 206–211, IEEE, 2015.
- [237] R. Elawady, S. Barakat, H. El-Bakry, and N. Elrashidy, “Sentiment Analysis for Arabic and English Datasets,” *IJICIS*, vol. 15, no. 1, 2015.
- [238] R. Bouchlaghem, A. Elkhelifi, and R. Faiz, “A Machine Learning Approach for Classifying Sentiments in Arabic Tweets,” in *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, p. 24, ACM, 2016.
- [239] R. Bouchlaghem, A. Elkhelifi, and R. Faiz, “Sentiment Analysis in Arabic Twitter Posts Using Supervised Methods with Combined Features,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 320–334, Springer, 2016.

- [240] Q. A. Al-Radaideh and L. M. Twaiq, "Rough Set Theory for Arabic Sentiment Classification," in *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, pp. 559–564, IEEE, 2014.
- [241] Q. A. Al-Radaideh and G. Y. Al-Qudah, "Application of Rough Set-based Feature Selection for Arabic Sentiment Analysis," *Cognitive Computation*, vol. 9, no. 4, pp. 436–445, 2017.
- [242] S. Alhumoud, T. Albuhairei, and M. Altuwaijri, "Arabic Sentiment Analysis Using WEKA a Hybrid Learning Approach," in *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, pp. 402–408, IEEE, 2015.
- [243] S. Alhumoud, T. Albuhairei, and W. Alohaideb, "Hybrid Sentiment Analyser for Arabic Tweets Using R," in *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, pp. 417–424, IEEE, 2015.
- [244] B. Brahim, M. Touahria, and A. Tari, "Data and Text Mining Techniques for Classifying Arabic Tweet Polarity," *Journal of Digital Information Management*, vol. 14, no. 1, 2016.
- [245] K. M. Alomari, H. M. ElSherif, and K. Shaalan, "Arabic Tweets Sentimental Analysis Using Machine Learning," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 602–610, Springer, 2017.
- [246] A. Elouardighi, M. Maghfour, and H. Hammia, "Collecting and Processing Arabic Facebook Comments for Sentiment Analysis," in *International Conference on Model and Data Engineering*, pp. 262–274, Springer, 2017.
- [247] A. Elouardighi, H. Hammia, and M. Maghfour, "Collecting and Processing Multilingual Streaming Tweets for Sentiment Analysis," in *International Conference on Information Technology and Communication Systems*, pp. 10–23, Springer, 2017.
- [248] A.-S. Ghadeer, I. Aljarah, and H. Alsawalqah, "Enhancing the Arabic Sentiment Analysis Using Different Preprocessing Operators," *New Trends in Information Technology*, p. 113, 2017.
- [249] A. E.-D. A. Hamouda and F. E.-z. El-taher, "Sentiment Analyzer for Arabic Comments System," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 3, 2013.
- [250] N. Boudad, R. Faizi, R. O. H. Thami, and R. Chiheb, "Sentiment Classification OF Arabic Tweets: A Supervised Approach," *Journal of Mobile Multimedia*, vol. 13, no. 3&4, pp. 233–243, 2017.

- [251] S. Alotaibi and C. Anderson, “Word Clustering as a Feature for Arabic Sentiment Classification,” *IJ Education and Management Engineering*, pp. 1–13, 2017.
- [252] M. N. Al-Kabi, A. H. Gigieh, I. M. Alsmadi, H. A. Wahsheh, and M. M. Haidar, “Opinion Mining and Analysis for Arabic Language,” *IJACSA International Journal of Advanced Computer Science and Applications*, vol. 5, no. 5, pp. 181–195, 2014.
- [253] M. El-Masri, N. Altrabsheh, H. Mansour, and A. Ramsay, “A Web-based Tool for Arabic Sentiment Analysis,” *Procedia Computer Science*, vol. 117, pp. 38–45, 2017.
- [254] A. M. Azmi and S. M. Alzanin, “Aara’—a System for Mining the Polarity of Saudi Public Opinion through E-newspaper Comments,” *Journal of Information Science*, vol. 40, no. 3, pp. 398–410, 2014.
- [255] M. M. Itani, R. N. Zantout, L. Hamandi, and I. Elkabani, “Classifying Sentiment in Arabic Social Networks: Naive Search Versus Naive Bayes,” in *Advances in Computational Tools for Engineering Applications (ACTEA), 2012 2nd International Conference on*, pp. 192–197, IEEE, 2012.
- [256] S. Atia and K. Shaalan, “Increasing the Accuracy of Opinion Mining in Arabic,” in *Arabic Computational Linguistics (ACLing), 2015 First International Conference on*, pp. 106–113, IEEE, 2015.
- [257] A. Mahmoud and T. Elghazaly, “Using Twitter to Monitor Political Sentiment for Arabic Slang,” in *Intelligent Natural Language Processing: Trends and Applications*, pp. 53–66, Springer, 2018.
- [258] A. Mountassir, H. Benbrahim, and I. Berrada, “Addressing the Problem of Unbalanced Data Sets in Sentiment Analysis,” in *KDIR*, pp. 306–311, 2012.
- [259] A. Mountassir, H. Benbrahim, and I. Berrada, “An Empirical Study to Address the Problem of Unbalanced Data Sets in Sentiment Classification,” in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pp. 3298–3303, IEEE, 2012.
- [260] A. Mountassir, H. Benbrahim, and I. Berrada, “Some Methods to Address the Problem of Unbalanced Sentiment Classification in an Arabic Context,” in *Information Science and Technology (CIST), 2012 Colloquium in*, pp. 43–48, IEEE, 2012.
- [261] A. M. Mostafa, “An Automatic Lexicon with Exceptional-Negation Algorithm for Arabic Sentiments Using Supervised Classification,” *Journal of Theoretical & Applied Information Technology*, vol. 95, no. 15, 2017.

- [262] N. Omar, M. Albared, A. Q. Al-Shabi, and T. Al-Moslmi, “Ensemble of Classification Algorithms for Subjectivity and Sentiment Analysis of Arabic Customers’ Reviews,” *International Journal of Advancements in Computing Technology*, vol. 5, no. 14, p. 77, 2013.
- [263] M. Biltawi, G. Al-Naymat, and S. Tedmori, “Arabic Sentiment Classification: a Hybrid Approach,” in *New Trends in Computing Sciences (ICTCS), 2017 International Conference on*, pp. 104–108, IEEE, 2017.
- [264] R. T. Khasawneh, H. A. Wahsheh, I. M. Alsmadi, and M. N. Al-Kabi, “Arabic Sentiment Polarity Identification Using a Hybrid Approach,” in *Information and Communication Systems (ICICS), 2015 6th International Conference on*, pp. 148–153, IEEE, 2015.
- [265] A. Bayoudhi, H. Ghorbel, and L. H. Belguith, “Sentiment Classification of Arabic Documents: Experiments with Multi-type Features and Ensemble Algorithms,” in *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pp. 196–205, 2015.
- [266] S. Al-Azani and E.-S. M. El-Alfy, “Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text,” *Procedia Computer Science*, vol. 109, pp. 359–366, 2017.
- [267] R. M. Duwairi, “Sentiment Analysis for Dialectical Arabic,” in *Information and Communication Systems (ICICS), 2015 6th International Conference on*, pp. 166–170, IEEE, 2015.
- [268] A. Y. Al-Obaidi and V. W. Samawi, “Opinion Mining: Analysis of Comments Written in Arabic Colloquial,” in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2016.
- [269] A. M. El-Halees, “Arabic Opinion Mining Using Distributed Representations of Documents,” in *Information and Communication Technology (PICICT), 2017 Palestinian International Conference on*, pp. 28–33, IEEE, 2017.
- [270] A. A. Altowayan and L. Tao, “Word Embeddings for Arabic Sentiment Analysis,” in *Big Data (Big Data), 2016 IEEE International Conference on*, pp. 3820–3825, IEEE, 2016.
- [271] A. A. Altowayan and A. Elnagar, “Improving Arabic Sentiment Analysis with Sentiment-specific Embeddings,” in *Big Data (Big Data), 2017 IEEE International Conference on*, pp. 4314–4320, IEEE, 2017.
- [272] S. Al-Azani and E.-S. M. El-Alfy, “Combining Emojis with Arabic Textual Features for Sentiment Classification,” in *Information and Communication*

- Systems (ICICS), 2018 9th International Conference on*, pp. 139–144, IEEE, 2018.
- [273] M. Elrazzaz, S. Elbassuoni, K. Shaban, and C. Helwe, “Methodical Evaluation of Arabic Word Embeddings,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 454–458, 2017.
- [274] R. Duwairi and M. El-Orfali, “A Study of the Effects of Preprocessing Strategies on Sentiment Analysis for Arabic Text,” *Journal of Information Science*, vol. 40, no. 4, pp. 501–513, 2014.
- [275] F. S. Al-Anzi and D. AbuZeina, “A Micro-word Based Approach for Arabic Sentiment Analysis,” in *Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on*, pp. 910–914, IEEE, 2017.
- [276] A. Abbasi, H. Chen, and A. Salem, “Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums,” *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, p. 12, 2008.
- [277] N. Farra, E. Challita, R. A. Assi, and H. Hajj, “Sentence-level and Document-level Sentiment Mining for Arabic Texts,” in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pp. 1114–1119, IEEE, 2010.
- [278] E. Refaee and V. Rieser, “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*, p. 16, 2014.
- [279] R. Duwairi, R. Marji, N. Sha’ban, and S. Rushaidat, “Sentiment Analysis in Arabic Tweets,” in *Information and communication systems (icics), 2014 5th international conference on*, pp. 1–6, IEEE, 2014.
- [280] M. Abdul-Mageed, M. Diab, and S. Kübler, “SAMAR: Subjectivity and Sentiment Analysis for Arabic Social Media,” *Computer Speech & Language*, vol. 28, no. 1, pp. 20–37, 2014.
- [281] W. Cherif, A. Madani, and M. Kissi, “A New Modeling Approach for Arabic Opinion Mining Recognition,” in *2015 Intelligent Systems and Computer Vision (ISCV)*, pp. 1–6, IEEE, 2015.
- [282] W. Cherif, A. Madani, and M. Kissi, “Towards an Efficient Opinion Measurement in Arabic Comments,” *Procedia Computer Science*, vol. 73, pp. 122–129, 2015.

- [283] W. Cherif, A. Madani, and M. Kissi, “A Combination of Low-level Light Stemming and Support Vector Machines for the Classification of Arabic Opinions,” in *Intelligent Systems: Theories and Applications (SITA), 2016 11th International Conference on*, pp. 1–5, IEEE, 2016.
- [284] D. Abuaiadah, D. Rajendran, and M. Jarrar, “Clustering Arabic Tweets for Sentiment Analysis,” in *Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on*, pp. 449–456, IEEE, 2017.
- [285] M. N. Al-Kabi, I. M. Alsmadi, R. T. Khasawneh, and H. A. Wahsheh, “Evaluating Social Context in Arabic Opinion Mining,” *The International Arab Journal of Information Technology*, 2016.
- [286] S. Oraby, Y. El-Sonbaty, and M. A. El-Nasr, “Exploring the Effects of Word Roots for Arabic Sentiment Analysis,” in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 471–479, 2013.
- [287] M. Abdul-Mageed, “Modeling Arabic Subjectivity and Sentiment in Lexical Space,” *Information Processing & Management*, 2017.
- [288] M. Abdul-Mageed, “Not All Segments are Created Equal: Syntactically Motivated Sentiment Analysis in Lexical Space,” in *Proceedings of the Third Arabic Natural Language Processing Workshop*, pp. 147–156, 2017.
- [289] O. Al-Harbi, “Using objective Words in the Reviews to Improve the Colloquial Arabic Sentiment Analysis,” *International Journal on Natural Language Computing (IJNLC)*, 2017.
- [290] L. Abd-Elhamid, D. Elzanfaly, and A. S. Eldin, “Feature-based Sentiment Analysis in Online Arabic Reviews,” in *Computer Engineering & Systems (ICCES), 2016 11th International Conference on*, pp. 260–265, IEEE, 2016.
- [291] M. Al-Smadi, M. Al-Ayyoub, H. N. Al-Sarhan, and Y. Jararweh, “An Aspect-based Sentiment Analysis Approach to Evaluating Arabic News Affect on Readers,” *Journal of Universal Computer Science*, vol. 22, no. 5, pp. 630–649, 2016.
- [292] M. Al-Smadi, O. Qwasmeh, B. Talafha, M. Al-Ayyoub, Y. Jararweh, and E. Benkhelifa, “An Enhanced Framework for Aspect-based Sentiment Analysis of Hotels’ Reviews: Arabic Reviews Case Study,” in *Internet Technology and Secured Transactions (ICITST), 2016 11th International Conference for*, pp. 98–103, IEEE, 2016.

- [293] I. Touati, M. Graja, M. Ellouze, and L. H. Belguith, “Arabic Fine-Grained Opinion Categorization Using Discriminative Machine Learning Technique,” in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 104–113, Springer, 2016.
- [294] I. Touati, M. Graja, M. Ellouze, and L. H. Belguith, “Towards Arabic Semantic Opinion Mining: Identifying Opinion, Polarity and Intensity,” in *Proceedings of the Mediterranean Conference on Pattern Recognition and Artificial Intelligence*, pp. 131–136, ACM, 2016.
- [295] A. Nuseir, M. Al-Ayyoub, M. Al-Kabi, G. Kanaan, and R. Al-Shalabi, “Improved Hierarchical Classifiers for Multi-way Sentiment Analysis,” *International Arab Journal of Information Technology (IAJIT)*, vol. 14, 2017.
- [296] M. Al-Ayyoub, A. Nuseir, G. Kanaan, and R. Al-Shalabi, “Hierarchical Classifiers for Multi-way Sentiment Analysis of Arabic Reviews,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 2, pp. 531–539, 2016.
- [297] J. B. Salamah and A. Elkhelifi, “Microblogging Opinion Mining Approach for Kuwaiti Dialect,” in *The International Conference on Computing Technology and Information Management (ICCTIM2014)*, pp. 388–396, The Society of Digital Information and Wireless Communication, 2014.
- [298] M. Al-Kabi, N. M. Al-Qudah, I. Alsmadi, M. Dabour, and H. Wahsheh, “Arabic/English Sentiment Analysis: an Empirical Study,” in *The Fourth International Conference on Information and Communication Systems (ICICS 2013)*, pp. 23–25, 2013.
- [299] H. K. Aldayel and A. M. Azmi, “Arabic Tweets Sentiment Analysis—a Hybrid Scheme,” *Journal of Information Science*, vol. 42, no. 6, pp. 782–797, 2016.
- [300] S. S. Alotaibi and C. W. Anderson, “Extending the Knowledge of the Arabic Sentiment Classification Using a Foreign External Lexical Source,” *Int. J. Nat. Lang. Comput.*, vol. 5, no. 3, pp. 1–11, 2016.
- [301] L. Abd-Elhamid, D. Elzanfaly, and A. S. Eldin, “Arabic Feature-based Level Sentiment Analysis Using Lexicon-based Approach,” *Journal of Fundamental and Applied Sciences*, vol. 10, no. 4S, pp. 143–148, 2018.
- [302] A. Elnagar, “Investigation on Sentiment Analysis for Arabic Reviews,” in *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*, pp. 1–7, IEEE, 2016.

- [303] A. R. Hedar and M. Doss, “Mining Social Networks Arabic Slang Comments,” in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2013.
- [304] T. H. Soliman, M. Elmasry, A. Hedar, and M. Doss, “Sentiment Analysis of Arabic Slang Comments on Facebook,” *International Journal of Computers & Technology*, vol. 12, no. 5, pp. 3470–3478, 2014.
- [305] A. M. AlAsmar, *Feature Based Approach in Arabic Opinion Mining Using Ontology*. PhD thesis, The Islamic University–Gaza, 2016.
- [306] A. El-Halees and A. Al-Asmar, “Ontology Based Arabic Opinion Mining,” *Journal of Information & Knowledge Management*, vol. 16, no. 03, p. 1750028, 2017.
- [307] R. Speer, J. Chin, and C. Havasi, “ConceptNet 5.5: an Open Multilingual Graph of General Knowledge.,” in *AAAI*, pp. 4444–4451, 2017.
- [308] F. Lazhar and T. G. Yamina, “Identification of Opinions in Arabic Texts Using Ontologies,” in *In Workshop on Ubiquitous Data Mining*, 2012, pp. 61–64, IEEE, 2012.
- [309] A. M. Alkadri and A. M. ElKorany, “Semantic Feature Based Arabic Opinion Mining Using Ontology,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, pp. 577–583, 2016.
- [310] S. Tartir and I. Abdul-Nabi, “Semantic Sentiment Analysis in Arabic Social Media,” *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 2, pp. 229–233, 2017.
- [311] S. Alowaidi, M. Saleh, and O. Abulnaja, “Semantic Sentiment Analysis of Arabic Texts,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 2, pp. 256–262, 2017.
- [312] A. Bani-Hani, M. Majdalawieh, and F. Obeidat, “The Creation of an Arabic Emotion Ontology Based on E-Motive,” *Procedia Computer Science*, vol. 109, pp. 1053–1059, 2017.
- [313] E. Refaee and V. Rieser, “Can We Read Emotions from a Smiley Face? Emoticon-based Distant Supervision for Subjectivity and Sentiment Analysis of Arabic Twitter Feeds,” in *5th International Workshop on Emotion, Social Signals, Sentiment and Linked Open Data, LREC*, 2014.
- [314] E. Refaee and V. Rieser, “Evaluating Distant Supervision for Subjectivity and Sentiment Analysis on Arabic Twitter Feeds,” in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pp. 174–179, 2014.

- [315] S. Al-Osaimi and K. M. Badruddin, “Role of Emotion Icons in Sentiment Classification of Arabic Tweets,” in *Proceedings of the 6th international conference on management of emergent digital ecosystems*, pp. 167–171, ACM, 2014.
- [316] S. Rizkallah, A. Atiya, H. E. Mahgoub, and M. Heragy, “Dialect Versus MSA Sentiment Analysis,” in *International Conference on Advanced Machine Learning Technologies and Applications*, pp. 605–613, Springer, 2018.
- [317] W. A. Al-Harbi and A. Emam, “Effect of Saudi Dialect Preprocessing on Arabic Sentiment Analysis,” *International Journal of Advanced Computer Technology (IJACT)*, 2015.
- [318] M. Mustafa, A. S. AlSamahi, and A. Hamouda, “New Avenues in Arabic Sentiment Analysis,” *International Journal of Scientific & Engineering Research*, pp. 907–915, 2017.
- [319] N. El-Naggar, Y. El-Sonbaty, and M. A. El-Nasr, “Sentiment Analysis of Modern Standard Arabic and Egyptian Dialectal Arabic Tweets,” in *Computing Conference, 2017*, pp. 880–887, IEEE, 2017.
- [320] G. Balikas and M.-R. Amini, “TwiSE at SemEval-2016 Task 4: Twitter Sentiment Classification,” *arXiv preprint arXiv:1606.04351*, 2016.
- [321] S. R. El-Beltagy, M. E. Kalamawy, and A. B. Soliman, “NileTMRG at SemEval-2017 Task 4: Arabic Sentiment Analysis,” *arXiv preprint arXiv:1710.08458*, 2017.
- [322] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” in *Advances in neural information processing systems*, pp. 3111–3119, 2013.
- [323] M. Jabreel and A. Moreno, “SiTAKA at SemEval-2017 Task 4: Sentiment Analysis in Twitter Based on a Rich Set of Features,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 694–699, 2017.
- [324] S. Miranda-Jiménez, M. Graff, E. S. Tellez, and D. Moctezuma, “INGEOTEC at SemEval 2017 Task 4: a B4MSA Ensemble Based on Genetic Programming for Twitter Sentiment Analysis,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 771–776, 2017.
- [325] E. S. Tellez, S. Miranda-Jiménez, M. Graff, D. Moctezuma, R. R. Suárez, and O. S. Siordia, “A Simple Approach to Multilingual Polarity

- Classification in Twitter,” *Pattern Recognition Letters*, vol. 94, pp. 68–74, 2017.
- [326] M. Graff, E. S. Tellez, S. Miranda-Jiménez, and H. J. Escalante, “Evodag: a Semantic Genetic Programming Python Library,” in *Power, Electronics and Computing (ROPEC), 2016 IEEE International Autumn Meeting on*, pp. 1–6, IEEE, 2016.
- [327] M. Graff, E. S. Tellez, H. J. Escalante, and S. Miranda-Jiménez, “Semantic Genetic Programming for Sentiment Analysis,” in *NEO 2015*, pp. 43–65, Springer, 2017.
- [328] S. Ismail, A. Alsammak, and T. Elshishtawy, “A Generic Approach for Extracting Aspects and Opinions of Arabic Reviews,” in *Proceedings of the 10th International Conference on Informatics and Systems*, pp. 173–179, ACM, 2016.
- [329] M. Mataoui, T. E. B. Hacine, I. Tellache, A. Bakhtouchi, and O. Zelmati, “A New Syntax-based Aspect Detection Approach for Sentiment Analysis in Arabic Reviews,” in *Natural Language and Speech Processing (ICNLSP), 2018 2nd International Conference on*, pp. 1–6, IEEE, 2018.
- [330] M. A. Ibrahim and N. Salim, “Aspect Oriented Sentiment Analysis Model of Arabic Tweets,” *International Journal of Computer Science Trends and Technology (IJCTST)*, 2016.
- [331] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, “Deep Recurrent Neural Network vs. Support Vector Machine for Aspect-based Sentiment Analysis of Arabic Hotels’ Reviews,” *Journal of Computational Science*, 2017.
- [332] M. Al-Smadi, M. Al-Ayyoub, Y. Jararweh, and O. Qawasmeh, “Enhancing Aspect-based Sentiment Analysis of Arabic Hotels’ Reviews Using Morphological, Syntactic and Semantic Features,” *Information Processing & Management*, 2018.
- [333] N. Farra and K. McKeown, “Smarties: Sentiment Models for Arabic Target Entities,” *arXiv preprint arXiv:1701.03434*, 2017.
- [334] T. Zarra, R. Chiheb, R. Moumen, R. Faizi, and A. E. Afia, “Topic and Sentiment Model Applied to the Colloquial Arabic: a Case Study of Maghrebi Arabic,” in *Proceedings of the 2017 International Conference on Smart Digital Environment*, pp. 174–181, ACM, 2017.
- [335] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global Vectors for Word Representation,” in *EMNLP*, vol. 14, pp. 1532–1543, 2014.

- [336] R. Collobert and J. Weston, “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning,” in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, ACM, 2008.
- [337] M. Al-Ayyoub, A. Nuseir, K. Alsmearat, Y. Jararweh, and B. Gupta, “Deep Learning for Arabic NLP: A survey,” *Journal of computational science*, 2017.
- [338] A. Tamchyna and K. Veselovská, “Ufal at Semeval-2016 Task 5: Recurrent Neural Networks for Sentence Classification,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 367–371, 2016.
- [339] A. El-Kilany, A. Azzam, and S. R. El-Beltagy, “Using Deep Neural Networks for Extracting Sentiment Targets in Arabic Tweets,” in *Intelligent Natural Language Processing: Trends and Applications*, pp. 3–15, Springer, 2018.
- [340] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A Convolutional Neural Network for Modelling Sentences,” *arXiv preprint arXiv:1404.2188*, 2014.
- [341] D. Tang, B. Qin, and T. Liu, “Document Modeling with Gated Recurrent Neural Network for Sentiment Classification,” in *EMNLP*, pp. 1422–1432, 2015.
- [342] R. Baly, R. Hobeica, H. Hajj, W. El-Hajj, K. B. Shaban, and A. Al-Sallab, “A Meta-Framework for Modeling the Human Reading Process in Sentiment Analysis,” *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 1, p. 7, 2016.
- [343] A. A. Al Sallab, R. Baly, G. Badaro, H. Hajj, W. El Hajj, and K. B. Shaban, “Deep Learning Models for Sentiment Analysis in Arabic,” in *ANLP Workshop*, vol. 9, 2015.
- [344] N. Abdelhade, T. H. A. Soliman, and H. M. Ibrahim, “Detecting Twitter Users’ Opinions of Arabic Comments During Various Time Episodes via Deep Neural Network,” in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 232–246, Springer, 2017.
- [345] N. Habash and F. Sadat, “Arabic Preprocessing Schemes for Statistical Machine Translation,” in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pp. 49–52, Association for Computational Linguistics, 2006.

- [346] R. Baly, H. Hajj, N. Habash, K. B. Shaban, and W. El-Hajj, “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)*, vol. 16, no. 4, p. 23, 2017.
- [347] A. Dahou, S. Xiong, J. Zhou, M. H. Haddoud, and P. Duan, “Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2418–2427, 2016.
- [348] A. M. Alayba, V. Palade, M. England, and R. Iqbal, “Improving Sentiment Analysis in Arabic Using Word Representation,” *arXiv preprint arXiv:1803.00124*, 2018.
- [349] M. Gridach, H. Haddad, and H. Mulki, “Empirical Evaluation of Word Representations on Arabic Sentiment Analysis,” in *International Conference on Arabic Language Processing*, pp. 147–158, Springer, 2017.
- [350] A. M. Alayba, V. Palade, M. England, and R. Iqbal, “Arabic Language Sentiment Analysis on Health Services,” in *Arabic Script Analysis and Recognition (ASAR), 2017 1st International Workshop on*, pp. 114–118, IEEE, 2017.
- [351] A. M. Alayba, V. Palade, M. England, and R. Iqbal, “A Combined CNN and LSTM Model for Arabic Sentiment Analysis,” in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pp. 179–191, Springer, 2018.
- [352] S. Ruder, P. Ghaffari, and J. G. Breslin, “A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis,” *arXiv preprint arXiv:1609.02745*, 2016.
- [353] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, *et al.*, “SemEval-2016 Task 5: Aspect Based Sentiment Analysis,” in *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pp. 19–30, 2016.
- [354] S. Ruder, P. Ghaffari, and J. G. Breslin, “Insight-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis,” *arXiv preprint arXiv:1609.02748*, 2016.

- [355] L.-s. YU and S. AL BAADANI, “A Sentiment Analysis Approach Based on Arabic Social Media Platforms,” *DEStech Transactions on Engineering and Technology Research*, no. icmeit, 2018.
- [356] S. Al-Azani and E.-S. El-Alfy, “Emojis-based Sentiment Classification of Arabic Microblogs Using Deep Recurrent Neural Networks,” in *Computing Sciences and Engineering (ICCSE), 2018 International Conference on*, pp. 1–6, IEEE, 2018.
- [357] S. Al-Azani and E.-S. M. El-Alfy, “Hybrid Deep Learning for Sentiment Polarity Determination of Arabic Microblogs,” in *International Conference on Neural Information Processing*, pp. 491–500, Springer, 2017.
- [358] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, and Y. Jararweh, “Using Long Short-term Memory Deep Neural Networks for Aspect-based Sentiment Analysis of Arabic Reviews,” *International Journal of Machine Learning and Cybernetics*, pp. 1–13, 2018.
- [359] A. Barhoumi, Y. E. C. Aloulou, and L. H. Belguith, “Document embeddings for Arabic Sentiment Analysis,” *Language Processing and Knowledge Management*, 2017.
- [360] M. Abdullah and S. Shaikh, “TeamUNCC at SemEval-2018 Task 1: Emotion Detection in English and Arabic Tweets using Deep Learning,” in *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 350–357, 2018.
- [361] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, “Semeval-2018 task 1: Affect in tweets,” in *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 1–17, 2018.
- [362] J.-A. González, F. Pla, and L.-F. Hurtado, “ELiRF-UPV at SemEval-2017 Task 4: Sentiment Analysis Using Deep Learning,” in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 723–727, 2017.
- [363] S. M. Mohammad, “Sentiment Analysis: Detecting Valence, Emotions, and Other Affectual States from Text,” *Emotion measurement*, pp. 201–238, 2015.
- [364] M. Araujo, J. Reis, A. Pereira, and F. Benevenuto, “An Evaluation of Machine Translation for Multilingual Sentence-level Sentiment Analysis,” in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 1140–1145, ACM, 2016.

- [365] E. Refaee and V. Rieser, “Benchmarking Machine Translated Sentiment Analysis for Arabic Tweets,” in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 71–78, 2015.
- [366] M. Rushdi-Saleh, M. T. Martín-Valdivia, L. A. Ureña-López, and J. M. Perea-Ortega, “Bilingual Experiments with an Arabic-English Corpus for Opinion Mining,” in *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pp. 740–745, 2011.
- [367] M. Bautin, L. Vijayarenu, and S. Skiena, “International Sentiment Analysis for News and Blogs.,” in *ICWSM*, 2008.
- [368] C. Banea, R. Mihalcea, and J. Wiebe, “Multilingual Subjectivity: are More Languages Better?,” in *Proceedings of the 23rd international conference on computational linguistics*, pp. 28–36, Association for Computational Linguistics, 2010.
- [369] A. M. Rabab’ah, M. Al-Ayyoub, Y. Jararweh, and M. N. Al-Kabi, “Evaluating Sentiment Strength for Arabic Sentiment Analysis,” in *Computer Science and Information Technology (CSIT), 2016 7th International Conference on*, pp. 1–6, IEEE, 2016.
- [370] R. T. Khasawneh, H. A. Wahsheh, M. N. Al-Kabi, and I. M. Alsmadi, “Sentiment Analysis of Arabic Social Media Content: a Comparative Study,” in *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*, pp. 101–106, IEEE, 2013.
- [371] S. R. El-Beltagy, T. Khalil, A. Halaby, and M. Hammad, “Combining Lexical Features and a Supervised Learning Approach for Arabic Sentiment Analysis,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 307–319, Springer, 2016.
- [372] D. Najjar and S. Mesfar, “Political Monitoring and Opinion Mining for Standard Arabic Texts,” in *NooJ’s conference*, 2014.
- [373] D. Najjar and S. Mesfar, “Opinion Mining and Sentiment Analysis for Arabic On-line Texts: Application on the Political Domain,” *International Journal of Speech Technology*, pp. 1–11, 2017.
- [374] T. Elghazaly, A. Mahmoud, and H. A. Hefny, “Political Sentiment Analysis Using Twitter Data,” in *Proceedings of the International Conference on Internet of things and Cloud Computing*, p. 11, ACM, 2016.
- [375] R. Abooraig, A. Alwajeih, M. Al-Ayyoub, and I. Hmeidi, “On the Automatic Categorization of Arabic Articles Based on their

- Political Orientation,” in *Third International Conference on Informatics Engineering and Information Science (ICIEIS2014)*, 2014.
- [376] M. Koppel, N. Akiva, E. Alshech, and K. Bar, “Automatically Classifying Documents by Ideological and Organizational Affiliation,” in *Intelligence and Security Informatics, 2009. ISI’09. IEEE International Conference on*, pp. 176–178, IEEE, 2009.
- [377] S. B. Hamouda and J. Akaichi, “Social Networks Text Mining for Sentiment Classification: the Case of Facebook Statuses Updates in the Arabic Spring Era,” *International Journal of Application or Innovation in Engineering & Management (IJAIEEM)*, vol. 2, no. 5, pp. 470–478, 2013.
- [378] K. Alsmearat, M. Shehab, M. Al-Ayyoub, R. Al-Shalabi, and G. Kanaan, “Emotion Analysis of Arabic Articles and its Impact on Identifying the Author’s Gender,” in *Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of*, pp. 1–6, IEEE, 2015.
- [379] A. Abu-Jbara, B. King, M. Diab, and D. Radev, “Identifying Opinion Subgroups in Arabic Online Discussions,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 829–835, 2013.
- [380] M. T. Khan and S. Khalid, “Sentiment Analysis for Health Care,” in *Big Data: Concepts, Methodologies, Tools, and Applications*, pp. 676–689, IGI Global, 2016.
- [381] N. Abdulla, *Towards Building a Sentiment Analysis Tool for Colloquial and Modern Standard Arabic Reviews*. PhD thesis, Master’s thesis. Computer Science Department, Jordan University of Science and Technology, Irbid, Jordan, 2014.
- [382] H. Wang, A. Hanafy, M. Bahgat, S. Noeman, O. S. Emam, and V. R. Bommireddipalli, “A System for Extracting Sentiment from Large-scale Arabic Social Data,” in *Arabic Computational Linguistics (ACLing), 2015 First International Conference on*, pp. 71–77, IEEE, 2015.
- [383] N. F. B. Hathlian and A. M. Hafezs, “Sentiment-Subjective Analysis Framework for Arabic Social Media Posts,” in *Information Technology (Big Data Analysis)(KACSTIT), Saudi International Conference on*, pp. 1–6, IEEE, 2016.
- [384] Y. Almas and K. Ahmad, “A Note on Extracting Sentiments in Financial News in English, Arabic & Urdu,” in *The Second Workshop on Computational Approaches to Arabic Script-based Languages*, pp. 1–12, 2007.

- [385] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng, “News Impact on Stock Price Return via Sentiment Analysis,” *Knowledge-Based Systems*, vol. 69, pp. 14–23, 2014.
- [386] H. Al-Hajieh, K. Redhead, and T. Rodgers, “Investor Sentiment and Calendar Anomaly Effects: a Case Study of the Impact of Ramadan on Islamic Middle Eastern Markets,” *Research in International Business and Finance*, vol. 25, no. 3, pp. 345–356, 2011.
- [387] H. Al-Rubaiee, R. Qiu, and D. Li, “Identifying Mubasher Software Products through Sentiment Analysis of Arabic Tweets,” in *Industrial Informatics and Computer Systems (CIICS), 2016 International Conference on*, pp. 1–6, IEEE, 2016.
- [388] A.-R. Hamed, R. Qiu, and D. Li, “Analysis of the Relationship between Saudi Twitter Posts and the Saudi Stock Market,” in *Intelligent Computing and Information Systems (ICICIS), 2015 IEEE Seventh International Conference on*, pp. 660–665, IEEE, 2015.
- [389] A.-R. Hamed, R. Qiu, K. Alomar, and D. Li, “Techniques for Improving the Labelling Process of Sentiment Analysis in the Saudi Stock Market,” *International Journal of Advanced Computer Technology (IJACT)*, 2018.
- [390] G. A. A. J. Alkubaisi, S. S. Kamaruddin, and H. Husni, “Stock Market Classification Model Using Sentiment Analysis on Twitter Based on Hybrid Naive Bayes Classifiers,” *Computer and Information Science*, vol. 11, no. 1, p. 52, 2018.
- [391] M. Alshahrani, F. Zhu, A. Sameh, L. Zheng, and S. Mumtaz, “Evaluating the Influence of Twitter on the Saudi Arabian Stock Market Indicators,” in *5th International Symposium on Data Mining Applications*, pp. 113–132, Springer, 2018.
- [392] K. AlKhatib, A. Rabab’ah, M. Al-Ayyoub, and Y. Jararweh, “On the Use of Arabic Tweets to Predict Stock Market Changes in the Arab World,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 7, no. 5, pp. 560–566, 2016.
- [393] A. Ahmed, S. Toral, and K. Shaalan, “Agent Productivity Measurement in Call Center Using Machine Learning,” in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 160–169, Springer, 2016.
- [394] L. Rahamatallah, E. Abuelyaman, and W. Mukhtar, “Constructing Opinion Mining Model of Sudanese Telecom Products,”

- [395] L. Almuqren and A. I. Cristea, “Framework for Sentiment Analysis of Arabic Text,” in *Proceedings of the 27th ACM Conference on Hypertext and Social Media*, pp. 315–317, ACM, 2016.
- [396] A. M. Qamar, S. A. Alsuhibany, and S. S. Ahmed, “Sentiment Classification of Twitter Data Belonging to Saudi Arabian Telecommunication Companies,” *International Journal of Advanced Computer Science and Applications (IJACS)*, vol. 1, no. 8, pp. 395–401, 2017.
- [397] H. Najadat, A. Al-Abdi, and Y. Sayaheen, “Model-based Sentiment Analysis of Customer Satisfaction for the Jordanian Telecommunication Companies,” in *Information and Communication Systems (ICICS), 2018 9th International Conference on*, pp. 233–237, IEEE, 2018.
- [398] H. Al-Rubaiee, R. Qiu, K. Alomar, and D. Li, “Sentiment Analysis of Arabic Tweets in e-Learning,” *Journal of Computer Science*, vol. 12, pp. 553–563, Jan 2016.
- [399] A. Eese and N. Omar, “A Hybrid Method for Arabic Educational Sentiment Analysis,” *Journal of Applied Sciences*, vol. 16, no. 5, pp. 216–222, 2016.
- [400] A. El-Halees, “Mining Changes of Opinions Expressed by Students to Improve Course Evaluation,” 2014.
- [401] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft, “Recommending Social Events from Mobile Phone Location Data,” in *2010 IEEE international conference on data mining*, pp. 971–976, IEEE, 2010.
- [402] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong, “Personalized Travel Package Recommendation,” in *2011 IEEE 11th International Conference on Data Mining*, pp. 407–416, IEEE, 2011.
- [403] Q. Yang, J. Fan, J. Wang, and L. Zhou, “Personalizing Web Page Recommendation via Collaborative Filtering and Topic-aware Markov Model,” in *2010 IEEE International Conference on Data Mining*, pp. 1145–1150, IEEE, 2010.
- [404] X. Jin, S. Spangler, Y. Chen, K. Cai, R. Ma, L. Zhang, X. Wu, and J. Han, “Patent Maintenance Recommendation with Patent Information Network Model,” in *2011 IEEE 11th International Conference on Data Mining*, pp. 280–289, IEEE, 2011.
- [405] J. Tang, S. Wu, J. Sun, and H. Su, “Cross-domain Collaboration Recommendation,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1285–1293, ACM, 2012.

- [406] B. Abdollahi and O. Nasraoui, “Transparency in Fair Machine Learning: the Case of Explainable Recommender systems,” in *Human and Machine Learning*, pp. 21–35, Springer, 2018.
- [407] P. Kouki, J. Schaffer, J. Pujara, J. O’Donovan, and L. Getoor, “Personalized Explanations for Hybrid Recommender Systems,” in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pp. 379–390, 2019.
- [408] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep Learning Based Recommender System: a Survey and New Perspectives,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.
- [409] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl, *et al.*, “Item-based Collaborative Filtering Recommendation Algorithms,” *Www*, vol. 1, pp. 285–295, 2001.
- [410] T. Hofmann, “Latent Semantic Models for Collaborative Filtering,” *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 89–115, 2004.
- [411] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, “A New User Similarity Model to Improve the Accuracy of Collaborative Filtering,” *Knowledge-Based Systems*, vol. 56, pp. 156–166, 2014.
- [412] A. Umyarov and A. Tuzhilin, “Improving Collaborative Filtering Recommendations Using External Data,” in *2008 Eighth IEEE International Conference on Data Mining*, pp. 618–627, IEEE, 2008.
- [413] D. D. Lee and H. S. Seung, “Learning the Parts of Objects by Non-negative Matrix Factorization,” *Nature*, vol. 401, no. 6755, p. 788, 1999.
- [414] G. Chen, F. Wang, and C. Zhang, “Collaborative Filtering Using Orthogonal Nonnegative Matrix Tri-factorization,” *Information Processing & Management*, vol. 45, no. 3, pp. 368–379, 2009.
- [415] D. D. Lee and H. S. Seung, “Algorithms for Non-negative Matrix Factorization,” in *Advances in neural information processing systems*, pp. 556–562, 2001.
- [416] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, “Algorithms and Applications for Approximate Nonnegative Matrix Factorization,” *Computational statistics & data analysis*, vol. 52, no. 1, pp. 155–173, 2007.

- [417] J. Kim, Y. He, and H. Park, “Algorithms for Nonnegative Matrix and Tensor Factorizations: a Unified View Based on Nlock Coordinate Descent Framework,” *Journal of Global Optimization*, vol. 58, no. 2, pp. 285–319, 2014.
- [418] C. H. Ding, T. Li, and M. I. Jordan, “Convex and Semi-nonnegative Matrix Factorizations,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 45–55, 2008.
- [419] P. Pirasteh, D. Hwang, and J. J. Jung, “Exploiting Matrix Factorization to Asymmetric User Similarities in Recommendation Systems,” *Knowledge-Based Systems*, vol. 83, pp. 51–57, 2015.
- [420] B. Ju, Y. Qian, and M. Ye, “Collaborative Filtering Algorithm Based on Structured Projective Nonnegative Matrix Factorization,” *Journal of Zhejiang University: Engineering Science*, vol. 49, no. 7, pp. 1319–1325, 2015.
- [421] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, “An Efficient Non-negative Matrix-factorization-based Approach to Collaborative Filtering for Recommender Systems,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1273–1284, 2014.
- [422] Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” *Computer*, no. 8, pp. 30–37, 2009.
- [423] Y. Koren and R. Bell, “Advances in Collaborative Filtering,” in *Recommender systems handbook*, pp. 77–118, Springer, 2015.
- [424] A. Gogna and A. Majumdar, “Blind Compressive Sensing Framework for Collaborative Filtering,” *arXiv preprint arXiv:1505.01621*, 2015.
- [425] M. Jamali and M. Ester, “Mining Social Networks for Recommendation,” *Tutorial of ICDM*, vol. 11, 2011.
- [426] J. Basiri, A. Shakery, B. Moshiri, and M. Z. Hayat, “Alleviating the Cold-start Problem of Recommender Systems Using a New Hybrid Approach,” in *2010 5th International Symposium on Telecommunications*, pp. 962–967, IEEE, 2010.
- [427] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, “Combing Content-based and Collaborative Filters in an Online Newspaper,” 1999.
- [428] M. Hornick and P. Tamayo, “Extending Recommender Systems for Disjoint User/item Sets: the Conference Recommendation Problem,” *IEEE*

- Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1478–1490, 2012.
- [429] J. Konstan and J. Riedl, “Recommended for You,” *Spectrum, IEEE*, vol. 49, pp. 54–61, 10 2012.
- [430] P. Resnick and H. R. Varian, “Recommender Systems,” *Communications of the ACM*, vol. 40, no. 3, pp. 56–59, 1997.
- [431] F. Armknecht and T. Strufe, “An Efficient Distributed Privacy-preserving Recommendation System,” in *2011 The 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop*, pp. 65–70, IEEE, 2011.
- [432] J. Oh, S. Park, H. Yu, M. Song, and S.-T. Park, “Novel Recommendation Based on Personal Popularity Tendency,” in *2011 IEEE 11th International Conference on Data Mining*, pp. 507–516, IEEE, 2011.
- [433] P. Karampiperis, A. Koukourikos, and G. Stoitsis, “Collaborative Filtering Recommendation of Educational Content in Social Environments Utilizing Sentiment Analysis Techniques,” in *Recommender Systems for Technology Enhanced Learning*, pp. 3–23, Springer, 2014.
- [434] G. Ganu, N. Elhadad, and A. Marian, “Beyond the Stars: Improving Rating Predictions Using Review Text Content,” in *WebDB*, vol. 9, pp. 1–6, Citeseer, 2009.
- [435] N. Pappas and A. Popescu-Belis, “Sentiment Analysis of User Comments for One-class Collaborative Filtering over Ted Talks,” in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 773–776, ACM, 2013.
- [436] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, “Introduction to WordNet: an On-line Lexical Database,” *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.
- [437] M. Maamouri, D. Graff, B. Bouziri, S. Krouna, and S. Kulick, “LDC Standard Arabic Morphological Analyzer (SAMA) v. 3.1,” *LDC Catalog No. LDC2010L01. ISBN*, pp. 1–58563, 2010.
- [438] W. S. Salloum and N. Y. Habash, “A Modern Standard Arabic Closed-Class Word List,” 2012.
- [439] T. Buckwalter, “Buckwalter {Arabic} Morphological Analyzer Version 1.0,” 2002.

- [440] C. Onyibe and N. Habash, “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)*, pp. 670–674, 2017.
- [441] A. El Kholly and N. Habash, “Orthographic and Morphological Processing for English–Arabic Statistical Machine Translation,” *Machine Translation*, vol. 26, no. 1-2, pp. 25–45, 2012.
- [442] P. Jain, P. Netrapalli, and S. Sanghavi, “Low-rank Matrix Completion Using Alternating Minimization,” in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674, ACM, 2013.
- [443] R. R. Coifman and M. Gavish, “Harmonic Analysis of Digital Data Bases,” in *Wavelets and Multiscale analysis*, pp. 161–197, Springer, 2011.
- [444] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, “Geometric Diffusions as a Tool for Harmonic Analysis and Structure Definition of Data: Diffusion Maps,” *Proceedings of the national academy of sciences*, vol. 102, no. 21, pp. 7426–7431, 2005.
- [445] L. W. Mackey, D. J. Weiss, and M. I. Jordan, “Mixed Membership Matrix Factorization,” in *ICML*, pp. 711–718, 2010.
- [446] K. Treerattanapitak and C. Jaruskulchai, “Exponential Fuzzy C-means for Collaborative Filtering,” *Journal of computer science and technology*, vol. 27, no. 3, pp. 567–576, 2012.