Sentiment Analysis of Egyptian Arabic in Social Media

Mohamed Abdalkader
Haverford College

Faculty Advisers
Deepak Kumar
Bryn Mawr College

Manar Darwish
Bryn Mawr College

April 25, 2014

Abstract[1]

Sentiment analysis is an emerging area of application fueled by the increase of public participation in online social media. Much work has been done on sentiment analysis in English while less work has been done on other languages like Mandarin and Arabic. Arabic is spoken by hundreds of millions of people in over twenty countries. Modern Standard Arabic (MSA) is used online mostly by newspapers and other official sources. However, social media and blogs used by individuals are typically in Dialect Arabic (DA). My Senior Thesis work has been focused on exploring ways to increase the accuracy of automated sentiment analysis in Egyptian Arabic through using the specific features of Arabic.

I found that the baseline algorithm makes the most mistakes in classifying tweets that carry a sentiment as neutral tweets. Using Minimum Edit Distance (MED) and ISRI Arabic stemmer, I was able to decrease the error of the baseline algorithm by 31% without having to add any new entries to the lexicon. My approach has allowed me to not only get over the challenge of different morphological forms but also misspelling and informal writing. While I cannot empirically compare it to results by other authors as I am using a different data set, my approach reaches an accuracy of 78% which has an improvement of 14.7% over the baseline.

[1]This work was supported in part by National Science Foundation through the Center for Science of Information (an NSF Science and Technology Center) award CCF-0939370 and the Haverford Marian E. Koshland Integrated Natural Sciences Center.

I have been extremely fortunate to have Professor Deepak Kumar and Professor Manar Darwish as my advisors. Without their direction and guidance, this thesis would not have been possible. I am also grateful to Professor Steven Lindell and Professor Samhaa Elbeltagy for their help and advice.

Table of Contents

1 Introduction

The most important asset for doing sentiment analysis is the sentiment lexicon. Sentiment lexicons have words and polarity scores assigned to them. However, a sentiment lexicon is not valuable if many of the words in the data set that carry a sentiment do not exist in the lexicon. Arabic is a very rich morphological language and each word may have more than a dozen morphological forms. Thus the rich morphology of Arabic becomes a big challenge for doing sentiment analysis, as words may exist in the data set in different forms from the ones existing in the lexicon and hence are not recognized by the algorithm. I worked on looking for solutions to overcoming the morphology issue by trying different computational linguistics algorithms.

Modern Standard Arabic (MSA) is used online mostly by newspapers and official sources. However, social media and blogs used by individuals are typically in Dialect Arabic (DA). With Egypt having the biggest online Arab community, Egyptian Arabic is the most widely used Arabic dialect on the internet. Thus I focus on Egyptian Arabic; however, conclusions reached can be generalized to any other dialect of Arabic. I used a corpus of hand-tagged tweets collected and tagged by the Center of Informatics Science at the Nile University to measure the performance. The dataset consists of 1679 tweets labeled as {*positive, negative, neutral*}. I used an Egyptian Arabic sentiment lexicon that has 4467 entries that are a mix of single words and composite terms [7].

Taking morphology into account I was able to increase the coverage of the lexicon I am using by 30.7%, which led to an improvement of 16.4% in the accuracy of the algorithm using the same lexicon without adding any new entries to the lexicon. I

conclude that morphology is a very important feature when doing sentiment analysis in Arabic and taking it into account can lead to a significant improvement in the accuracy of the polarity score given by sentiment analysis algorithms.

2 Background

Sentiment Analysis (SA) is the extraction of subjective information from text using computational techniques. It is an emerging area of application fueled by the increase of public participation in online social media. Sentiment analysis of people's writings in social media can serve as a measure of public sentiment about different topics. People share their opinions about politics, commercial products, and many other topics. The availability of these enormous amounts of opinions makes automated sentiment analysis valuable in many applications in business and government intelligence. Startups like ZEFR use Automated Sentiment Analysis to do brand tracking and management by analyzing comments on YouTube videos [16]. Other efforts have been made to analyze sentiment related to different stocks by analyzing texts from various blogs and websites [1].

Sentiment analysis could be performed at the document level, sentence level, or aspect based sentiment analysis. Document level sentiment analysis is the simplest form of sentiment analysis in which it is assumed that the document contains one opinion about a certain topic. Since a single document may contain multiple opinions about the topic, sentence level sentiment analysis becomes necessary for a more fine-grained analysis. It is assumed that the topic being discussed is known and that each sentence has only one opinion. Finally, the aspect based sentiment analysis is used when

the text analyzed contains opinions about more than one topic. For example, a car review can be: “The car’s engine is very good, the stereo is nice too but the seats are not comfortable.” In this case, an aspect-based analysis, will not give an overall score for the car but rather different scores for different aspects of the car.

2.2 Current work and approaches

There are two main approaches used for Sentiment Analysis. The first approach is the *semantic orientation* approach in which semantics of words is used to measure the subjectivity in the text. Since semantic orientation depends on the meanings of the word it is independent of the domain and thus the accuracy is reliable across different domains. A classical method of calculating semantic orientation of text is calculating the differences between the PMI (Point wise mutual information) of the text with two opposite words like “Good’ and “Bad” or “Excellent” and “Poor”. Kamps et al. reached 71% accuracy using this approach on the manually constructed lists of the General Inquirer [3][15]. They used WorldNet [14] to measure PMI by operationalizing Osgood’s *Evaluative Factor* (EVA) by defining a function that measures the relative distance of a word to the words “good” and “bad” in WorldNet graph. Then they it divided by the distance between “good” and “bad” to get a value in the range of [1,-1] .

$$EVA(word) = \frac{distance(word,'bad') - distance(word,'good')}{distance('good','bad')}$$

For example, the EVA for the word “honest” will get score of 1 as it is closer to the word “good” than it is to the word “bad”, thus it will be tagged as a positive word.

$$EVA('honest') = \frac{distance('honest','bad') - distance('honest','good')}{distance('good','bad')} = \frac{6-2}{4} = 1$$

The second approach is a *machine learning approach*. In its simplest form, sentiment analysis can be done as a basic classification problem between the two classes, {Positive, Negative}. For example, a model trained on product reviews will consider adjectives as features will tag adjectives like "great" positively as they will appear more often in positive reviews [2]. Classification algorithms like Maximum Entropy, Naïve Bayes and SVM can learn the classification given training data [1][21]. A model trained on annotated corpuses can consider features like: POS (parts of speech), word frequencies, and uniqueness to classify the text as positive or negative. Using machine learning to do sentiment analysis results in accuracy as high as 82% using SVM on movie reviews [4]. While machine-learning approaches can get better results than a semantic orientation approach, it is not reliable outside the training domain. Let's consider a model that was trained on a movie reviews dataset and used a unigram analysis to tag words as positive or negative. This model may end up tagging the word "scary" as positive while it is considered negative in most other domains. The positive tagging may come from the fact that the horror movies rated highly may use the word "scary" and thus the model will learn it as a positive word.

2.3 Arabic Corpuses

Labeled data sets are a very important component of building sentiment analysis systems. Realizing this, Abdul-Mageed *et al.* created AWATIF, a multi-genre, multi-dialect corpus for Arabic SSA [11]. AWATIF is extracted from three resources: Wikipedia user talk pages on different topics, the Penn Arabic Treebank [17], which is an existing collection of

news wire stories in different domains and conversation threads from web forums on seven different sites. Abdul-Mageed *et al.* labeled the corpus using both regular as well as crowd sourcing methods [11]. Another corpus was created by Elaranoty *et al*. They crawled 150 MB of Arabic news and manually annotated 1 MB of the corpus [10]. Three different people annotated the corpus. Majority voting was used when conflicts occurred. Finally, another small tagged dataset of approximately 500 Arabic tweets labeled by the tags {*Positive, Negative, Neutral}* was collected by Samhaa *et al* [5]. Having different tagged datasets is very important for sentiment analysis research so as to be able to evaluate different algorithms and approaches as well as training models if machine learning is to be used

2.4 Arabic Sentiment Lexicons

There are two approaches for creating sentiment lexicons, a *dictionary based approach* and a *corpus based approach*. The dictionary based approach starts with seed words of known positive or negative orientation that are manually collected. Next, synonyms are added to both sets and the process iterates until there are no more words left [2]. A more sophisticated approach was proposed by Kamps *et al.* using the distance between words in WordNet using the EVA formula explained before.

The Corpus based approach has been applied to two main scenarios. First, given a seed list of known sentiment words discover other sentiment words and their orientations from a domain corpus. The second is adapting a general-purpose sentiment lexicon to a new one using a domain corpus for sentiment analysis applications in this domain. The sentiment lexicon is the most crucial resource for sentiment analysis

research [2]. Thus the availability of these Arabic lexicons is a great asset and is important for the research on Arabic sentiment analysis to expand upon.

2.4.1 Egyptian Arabic Lexicon

Samhaa *et al.* blames the lack of accuracy of Sentiment Analysis on dialectical Arabic to various things including the unavailability of sentiment lexicons. Thus, they built a sentiment lexicon for Egyptian Arabic that contains 4467 words and composite phrases [5]. The lexicon was seeded by 380 words and then expanded based on the idea that positive words tend to appear with other positive adjectives when conjugated with "*and".* For example, if the lexicon was seeded with the word "honest" and if a sentence appeared expressing that: "Sami is honest and respectable." Then "*respectable"* would be identified as positive as well.

The lexicon was then manually revised. The lexicon was also assigned weights so as to increase the accuracy of the polarity scores calculated using the lexicon. The weights were assigned by extracting tweets (t) including word (w) in the lexicon. The tweets including only w and no other words from the lexicon were excluded. Then the numbers of tweets (O) with opposite words to w were counted. The weight was calculated using the formula:

$$weight(word) = \frac{t-o}{o}$$

2.4.2 Modern Standard Arabic Lexicon (Sifat)

Abdul-Mageed *et al.* manually created a polarity lexicon (Sifat) of 3982 words labeled with *{Positive, Negative, Neutral}* [7]*.* The adjectives in Sifat pertain to the newswire domain and were extracted from the first four parts of the Penn Arabic Treebank (PATB) [17]. The Lexicon is for Modern Standard Arabic unlike the one created by Samhaa *et al* [5] which is for Egyptian Arabic. Realizing the need for wider coverage, they expanded Sifat by translating three English Lexicons, Senti-Wordnet, YouTube Lexicon, and General Inquirer Lexicon [13][15][7]. They used the Google Translation API for translating the three lexicons. Their expanded lexicon includes 229,452 entries. However, the new expanded lexicon still under development is yet to be evaluated.

2.4.3 Arabic Senti-WordNet

SentiWordNet (SWN) is a lexical resource used for sentiment analysis [12]. The effort behind SWN comes from the realization that words have different meanings and thus may not always have the same polarity score. SWN assigns three scores to each word, *objectivity, positivity or negativity* score. The sum of these three scores always adds up to one. The main releases of SWN are SWN 1.0 and SWN 3.0 [12][13]. Alhazmi *et al*. have created an Arabic equivalent of SWN. They first started with upgrading Arabic WordNet (WN) 2.0 to 3.0. Then they mapped the English SWN 3.0 to the newly created Arabic WN 3.0. Finally they exclude all the entries in Arabic WN that do not exist in English SWN [9]. However, Abdul-Mageed *et al.* reported problems with coverage and quality in SWN 3.0 [6]. They suggest expanding it using Social Media Lexica [6].

2.5 SAMAR

Abdul-Mageed *et al.* present (SAMAR) a System for Subjectivity and Sentiment Analysis of Arabic Social Media [6]. SAMAR is an SVM-based system that uses two classifiers: subjectivity classifier and sentiment classifier. The subjectivity classifier determines whether a sentence is subjective or objective while the sentiment classifier determines whether it is positive or negative. They use SAMAR to approach four research questions: best representation of lexical information, the impact genre specific features have on performance, the usefulness of standard features and how to treat Arabic dialects. They reported accuracy using different combinations of features and compared it to the baseline. They considered the baseline tagging the entire dataset with the the majority class in the training data.

They used a corpus of data in both MSA and DA that is extracted from Twitter tweets, Web forums, chat and Wikipedia Talk pages. They used a manually created lexicon of 3982 adjectives labeled as {*Positive, Negative or Neutral*}. They used Genre specific features like UID (User ID) which has two values: {*Person, Organization}* and Gender with the values {*Male, Female, Unknown}.* The Morphological features were explored through replicating experiments using words as tokens or using their lemmas as well as POS (Parts of Speech) tagging of the words using both the RTS (*Reduced Tag Set*) and the ERTS (Extended Reduced Tag Set). Dialectical features were also included to use the values: {*DA, MSA}.* Finally, the standard features used were the PL (Polarity Lexicon) described above and adding the feature *UNIQUE* to low frequency words.

Results vary giving that no clear trend appears when morphology is used alone but accuracy increased when morphology is considered along with the POS tagging. Best results are reached when the lemma is used alone with the ERTS tag. Arabic is a very rich morphological language. For example, the word "شافها" means "*He saw her*." The lemma in this case would be "شاف". ERTS outperforming RTS imply the importance of using more detailed tags as ERTS has more tags than RTS. While they found that knowing whether the text is in MSA or DA has no difference, they found that SAMAR has higher accuracy for DA over MSA tweets with accuracy of 79% versus 65%. This is due to the fact that the majority of MSA tweets are from newspapers which tend to be more subtle with showing sentiment than normal users; however, using the UID, organization vs. individual, along with the other genre specific features improved the accuracy with 0.5% only on twitter data. Finally, they reach to the conclusion that each domain and task needs individualized solution. However, lemmatization is found to be important in all best approaches in Arabic.

3 Methodology

3.1 Data

I used a corpus of hand-tagged tweets collected and tagged by the Center of Informatics Science at the Nile University to measure the performance. The dataset consists of 1679 tweets labeled as {*positive, negative, neutral*}. I used an Egyptian Arabic sentiment lexicon that has 4467 entries that are a mix of single words and composite terms [7]. I chose to use this lexicon since its entries were extracted from tweets which will increase the performance given that the dataset consists of tweets as well.

3.2 Baseline Algorithm

I used a basic Semantic orientation algorithm that sums the polarity scores of each word and terms consisting of multiple words in the tweet as the baseline as described in Figure 1. The algorithm classifies tweets with a score over zero as positive, less than zero as negative, and zero as neutral. I used NLTK punctuation tokenizer to tokenize the tweet, which is more accurate than simply splitting the tweet into words by spaces. I used the Egyptian Arabic Sentiment Lexicon created by Samhaa *et al.* The Lexicon consists of both words and terms.

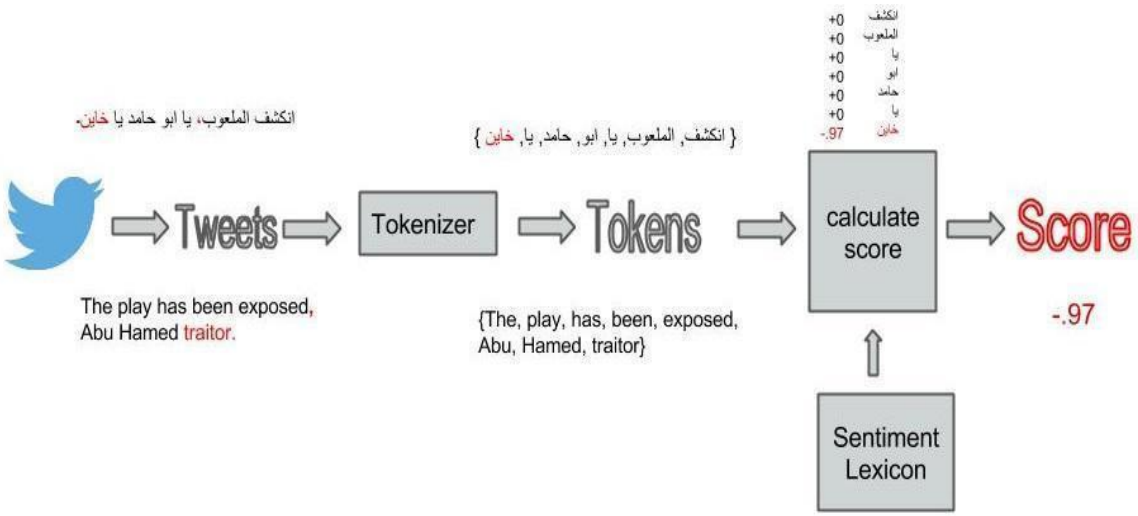

Figure 1

```
lookUpWordScore(word, lexicon):
     if word∈lexicon:
          return its score
     else:
          return 0

analyzeTweet(tweet):
     score←0
     tokens←tokenize(tweet)
     terms←getTerms(tweet)
     for token in tokens:
          tokenScore←lookUpWordScore(token)
          score←score + tokenScore
     for term in terms:
          termScore←lookUpWordScore(term)
          score←score + termScore
return score

classify(tweet):
     score←analyzeTweet(tweet)
     if  tweetScore >0:
          return positive
     else if tweetScore < 0:
          return negative
     else:
          return neutral
```

Algorithm 1

3.3 Experiments

Looking at the results from A*lgorithm 1* (Base Line) I found that the algorithm makes the most mistakes in classifying a tweet as neutral when it isn't. Looking at the output I found that the words in the tweets classified as neutral are in the lexicon; however, they exist in a different morphological form. Thus I experimented with different algorithms to overcome the challenge of the inability to recognize words existing in the lexicon.

The process, described in *Figure 1*, consists of three main phases. The first is tokenizing the tweet to words. Then looking up the words in the sentiment lexicon and finally assigning a score to each word and to overall tweet. I used the NLTK tokenizer in the baseline as well as in all the experiments. NLTK tokenizer proved more accurate than simply splitting the sentence to words where there are spaces as NLTK tokenizer takes into account punctuation and filters them properly. The work was focused on improving the lookup phase of the process and this led to modifying the final stage of assigning scores as well. I experimented using Minimum Edit Distance and ISRI stemmer to overcome the morphology challenge and increase the coverage of the algorithm. As we will see below, increasing the algorithm coverage naturally led to an increase in accuracy.

3.3.1 Minimum Edit Distance (MED)

The difference between morphological forms of words in Arabic is usually a change of one or two letters in a suffix or a prefix of the word. Thus, I explored using Minimum Edit Distance (MED) to overcome the morphological richness of Arabic. MED is an algorithm used to measure the similarity between two words. It counts the number of inserts and removes needed for one word to match the other. For example, the word “like” has an MED of 1 with the word “likes” as it only takes adding the “s” to “like” to be “likes”. I used the MED module provided by NLTK. I considered two words to have the same meaning if they have an MED below a certain threshold, γ. In order not to overweight score of words that make an exact match to an entry in the lexicon with

the words that I match through MED, I scale the score of words matched through MED with α, where α is less than one. I experimented with different values for α and γ.

```
lookupWordScore(word):
     if word∈lexicon:
          return its score
     for entry in lexcon:
          if MED(word, entry) < γ
               return word Score * α
     return 0
```

Algorithm 2

MED increased the coverage to almost 100%. Only one tweet out of the 1679 tweets was not assigned a score. However this was accompanied with a drastic decrease in accuracy. The accuracy drop is attributed to the false matches MED created. Many of the words were matched with words in the lexicon that have very different meanings. For example, the word "اعتمر" means "went to pilgrimage"; however it was matched to the word "اعتقل" which means "arrested" as they have an MED of 2. However the matched word carries a sentiment score of *-1* while the word in the tweet is neutral and does not carry any sentiment score.

An intuitive solution to the issue of false matches MED created was to normalize the MED with the length of the word to make sure the word is not completely changed. I subtracted normalized MED from one to get what I call match score to use to decide if two words should be considered the same. Using the following is the formula to calculate match score, accuracy increased to 77.6%:

$$MS(w1,\ w2)\ =\ 1 - \frac{MED(w1,\ w2)}{|w1|}$$

3.3.2 Jizr (Word Root)

Due to the richness of Arabic morphology a word can have many different forms. Adding all the forms of each entry in the lexicon will drastically increase the size of the lexicon. For example the adjective generous can have the following forms:

1. كريم/Kareem/Generous (Indefinite Masculine Singular)
2. كريمة/Kareema (Indefinite Feminine Singular)
3. الكريم/AlKareem (definite Masculine Singular)
4. الكريمة/Alkareema (definite Feminine Singular)
5. كرما/Korama' (Indefinite Masculine Plural)
6. الكرما/AlKorama' (definite Masculine Plural)
7. كريمات/Kareemat (Indefinite Feminine Plural)
8. الكريمات/AlKareemat (definite Feminine Plural)
9. أكرم/Akram (Comparative)
10. الأكرم/Alakram (Superlative)

I used the Information Science Research Institute (ISRI) stemmer [18], to get the root of each word in the tweet and compare it to the root of the entries in the lexicon. The stemmer gets the proper root for the word and if it fails to find it due to the complexity for some words it returns a stem a normalized form rather than returning the original unmodified form. For example the word "صلوات"/"prayers" should have the Jizr "صلو" but ISRI stemmer returns "صلا" instead which is wrong but close to it. To avoid overweighting the words that are found to be in the lexicon without using the stemmer, I multiply the score given to words recognized by the stemmer by α. So instead of simply looking up the word in the lexicon I use Algorithm 3.

```
lookupWordScore(word):
     if word∈lexicon:
          return its score
     for entry in lexcon:
          if stem(word) = stem(entry)
               return word Score * α
     return 0
```

Algorithm 3

As expected the coverage increased to 98% since getting the Jizr for each word overcame the morphology problem and recognized words in different morphological forms. For example, all of the words in tweet " الله يكرمك يا جميلة " were given a polarity score of *0* while the word "جميلة " has a positive score and is in the lexicon but in the masculine form: "جميل ". However when the Jizr of the word was used, both the word and the entry in the lexicon got the Jizr "جمل" and thus the algorithm assigned it a positive score of "جميل " and classified the tweet as a positive one.

3.3.3 Combining Approaches

To benefit from the increase of the accuracy of each algorithm I experimented with combining more than one algorithm together. I calculated the sentiment score in 4 stages. First, I check if the word is in the lexicon. If it is, I assign it the sentiment score in the lexicon. Second, I use Algorithm 3 to calculate the score based on the Jizr with a factor of 0.25. In the third stage, I use match score with $\gamma = 0.7$ and a factor of 0.25. Finally, I calculate the score of the composite terms through the function in Algorithm 1.

While using the Jizr gave slightly better results than using the match score, looking at the output, the two algorithms scored different tweets differently. Match score did not only act as a good heuristic to identify different morphological forms of the same words, but it also solved other problems. Using match score allowed the algorithm to recognize words that are misspelled. Given that the domain is social media, misspellings are common. For example the word جمييل meaning beautiful is misspelled as the vowel "ي" is doubled. This could be intentional to emphasize the word or unintentional misspelling, thus it was not recognized using neither the baseline algorithm nor using the Jizr as the ISRI stemmer returned the same word as the Jizr. However it has a match score of 0.8% with the word entry in the lexicon: "جميل". And thus it was recognized and classified correctly. Thus the combined approach scored better.

4 Results

4.1 MED

Using MED did not improve accuracy with any combination of α or γ. Table 2 shows the best accuracy reached with a $\gamma = 3$ compared to the baseline.

Table 1: N=1679

Experiment	False Neutrals	Coverage	Total Misses	Accuracy
Baseline	407	75.7%	553	68.2%
MED $\gamma = 3$ $\alpha = 0.25$	1	**99.9%**	653	**61.1%**

However using match score increased the accuracy. I used this with different values for γ and α. Using $\gamma = 0.5$ increased the coverage again to almost 100% with only 3 tweets classified as neutral; however, this time the accuracy witnesses a slight jump to 73.8% from 68.2%. Using $\gamma = 0.8$ increased the coverage score very slightly from 75.7% to 79.8%. This increase in coverage was accompanied by a similar slight increase in the accuracy score from 68.2% to 69.4%. This slight increase can be attributed to the failure to recognize that many more words than the basic algorithm. The best accuracy was reached with $\gamma = 0.7$. It reached an accuracy of 77.6% and a coverage score of 96.9%.

Match Score resulted in accuracy less than Jizr because it can cause false positives in which a word is mistakenly recognized to be in the lexicon when it is not. For example the word "جميل"/"Jamil" means beautiful while the word "جمل"/"jaml" means camel and they have a match score of 0.75, which is greater than γ and thus will be falsely recognized. However since the number of these cases is much smaller than the number of the cases in which the words are in fact another morphological form of the lexicon entry, the overall accuracy still increased. Table 2 shows best results reached using match score.

Table 2: N=1679

Experiment	False Neutrals	Coverage	Total Misses	Accuracy
Baseline	407	75.7%	553	68.2%
Match Score $\gamma = 0.5$ $\alpha = 0.25$	3	**99.9%**	653	73.8%
Match Score $\gamma = 0.7$ $\alpha = 0.25$	51	96.9%	439	**77.6%**

Match Score $\gamma = 0.8$ $\alpha = 0.25$	338	79.8%	513	69.4%

4.2 Jizr

I experimented with different values for α. While different values gave different results, all results showed a significant improvement over the baseline unlike MED and Match Score results that may not show an improvement over the baseline with different values for γ or α. The accuracy increased to 75% without reducing the score given at all (i.e. $\alpha = 0$) the best accuracy was reached with a α of 0.25 scoring 78.7%

Table 3:N=1679

Experiment	False Neutrals	Coverage	Total Misses	Accuracy
Baseline	407	75.7%	553	68.2%
Jizr $\alpha = 0$	33	**98%**	418	75.1%
Jizr $\alpha = 0.25$	33	**98%**	357	**78.7%**

4.3 Combining Approaches

Since Match Score solved the problem of misspelling while the Jizr approach did not but the Jizr approach proved more accurate identifying different morphological forms, using Jizr as a primary approach and match score as a secondary one increased the accuracy to 79.4% using $\gamma = 0.7$ and $\alpha = 0.25$ as shown in the table below.

Table 4: N=1679

Experiment	False Neutrals	Coverage	Total Misses	Accuracy
Baseline	407	75.7%	553	68.2%
Match score + Jizr $\gamma = 0.7$ $\alpha = 0.25$	15	**99%**	345	**79.4%**

4.6 Results Summary

Table 5 and Chart 1 shows the best results obtained from the different experiments. The best result was obtained using both Jizr and using Match Score if the stemmer could not match the word to any entry in the lexicon. This approach improved the accuracy by 16.4% and increased the coverage by 30.7%.

Table 5: N=1679

Experiment	False Neutrals	Coverage	Total Misses	Accuracy
Baseline	407	75.7%	553	68.2%
Match Score α *=0.25* γ *=0.**7***	51	96.9%	439	77.6%
Jizr α *=0.25*	33	98%	357	78.7%
Match score + Jizr γ *=0.7* α *= 0.25*	15	**99%**	345	**79.4%**

5 Conclusions

The experiments show that the richness of the Arabic morphology is very important to take into account when doing sentiment analysis on Arabic. Approaches that ignore morphology are missing a big piece of the puzzle that can improve the accuracy significantly. This also shows that more research needs to be done on Arabic morphology and ways to find words roots, as this will have strong impact on other research areas and applications that work with the Arabic Language.

The consistency of the increase of the accuracy when using the Jizr regardless of the value of α unlike the case in MED, proves that the increase in accuracy is not arbitrary and is attributed to taking into account morphology. On the other side Match score did not show a consistent increase of accuracy regardless of the values of α and γ. This shows that it can be good heuristics to use but it is not the optimal solution to the morphology problem.

In this research I only focused on improving the polarity score by solving the problem of morphology. However, there exists other work that tackled different problems in Arabic Sentiment Analysis. For example, Samhaa *et al*. have worked on detecting Arabic persons' names to factor them out of the polarity score, as Arabic names are usually adjectives. [19] Other efforts have been made by Abdul-Mageed *et al* in using the machine learning approach to do sentiment analysis in Arabic. Polarity score was among the features they used to train the model. More effort needs to be put into combining those different parts together to develop an Arabic Sentiment Analysis System that has higher accuracy.

Bibliography

[1] Feldman, R. "Techniques and Applications for Sentiment Analysis." *Communications of the ACM*, 2013.

[2] Liu, B. *Sentiment Analysis and Subjectivity.* Handbook of Natural Language Processing, Second Edition, 2010.

[3] Kamps, Jaap, Marten Marx, Robert J. Mokken, and Maarten de Rijke. "Using WordNet to Measure Semantic Orientations of Adjectives." *Language & Inference Technology Group ILLC, University of Amsterdam .*

[4] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan . "Thumbs up? Sentiment Classification using Machine Learning Techniques ." *Proceedings of EMNLP 2002*, 2002: 79-86.

[5] El-Beltagy, S., and A. Ali. "Open Issues in the Sentiment Analysis of Arabic Social Media: A Case Study." *Innovations in Information Technology*, 2013: 215-220.

[6] Kubler, S., M. Abdul-Mageed, and M. Diab. "SAMAR: A System for Subjectivity and Sentiment Analysis of Arabic Social Media." *Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysi*, 2012: 19–28.

[7] Abdul-Mageed, M., and M. Diab. "Toward Building a Large-Scale Arabic Sentiment Lexicon." *Proceedings of the 6th International Global Word-Net Conference*, 2012.

[8] Abdul-Mageed, M., D. Carndall, and M. Korayem. "Subjectivity and Sentiment Analysis of Arabic: A Survey."

[9] Alhazmi, S., W. Black, and J. McNaught. "Arabic SentiWordNet in Relation to SentiWordNet 3.0." *International Journal of Computational Linguistics*, 2013.

[10] Abdelrahman, S., M. Elaranoty, and A. Fahmy. "A Machine Learning Approach for Opinion Holding Extraction in Arabic Language." *International Journal of Artificial Intelligence and Applications*, 2012.

[11] Abdul-Mageed, M., and M. Diab. "AWATIF: A Multi-Genre Corpus for Modern Standard Arabic Subjectivity and Sentiment Analysis."

[12] Esuli, A., and F. Sebastiani. "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining." (Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche).

[13] Baccianella, S., A. Esuli, and and F. Sebastiani. "SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." (Istituto di Scienza e Tecnologie dell'Informazione).

[14] Fellbaum, C. "WordNet: An Electronic Lexical Database." (Cambridge, MA: MIT Press.) 1998.

[15] Stone, P. J., D. C. Dunphy, M. S. Smith, and D. M. Ogilvie. "The General Inquirer: a computer approach to content analysis." (MIT Press, Cambridge, MA.).

[16] *ZEFR.* http://zefr.com/ .

[17] *Penn Arabic Treebank*. http://www.ircs.upenn.edu/arabic/.

[18] Taghva, K., Elkoury R., and J. Coombs. "Arabic Stemming without a root dictionary." (Information Science Research Institute. University if Nevada) 2005.

[19] Samhaa R. El-Beltagy and Ahmed Rafea. A Corpus Based Approach for the Automatic Creation of Arabic Broken Plural Dictionaries, *Proceedings of 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2013)*. Samos-Greece 2013.

[20] *Egypt's internet economy could grow by 22% a year - study.* 2012. http://www.telecompaper.com/news/egypts-internet-economy-could-grow-by-22-a-year-study--912564.

[21] Cortes, C.; Vapnik, V. (1995). "Support-vector networks". Machine Learning 20