ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 12, June 2014

Insult Detection in Arabic On-line Commentary

Emad Mohamed

***Abstract*—we use a stochastic gradient descent classifier to detect insults in user-generated Arabic newspaper commentary. The language of the comments is a continuum of Modern Standard Arabic and colloquial Egyptian Arabic, and it deviates significantly from the standard orthographic and morphological norms. We use orthographic normalization and a stemmer trained on the Arabic Treebank and colloquial data with tf-idf, and we try a variety of learner settings in a 10-fold cross validation to achieve and f-score of 0.85. Precision and recall, however, vary considerably among different settings and per category. A list of insulting words did not prove helpful in the classification. Our results are better than those obtained on English in the Kaggle competition that inspired this work.**

***Index Terms*—Insult detection, Sentiment Analysis, Arabic NLP, Egyptian Arabic, Machine Learning, Computational Linguistics.**

I. INTRODUCTION

Word net defines insult as "a rude expression intended to offend or hurt". Rude expressions can be direct or in the form of asteism, aka back-handed compliments, which are deriding expressions in polite forms. In this paper, we try to automatically detect insulting comments posted in online forums in Arabic, especially on Egyptian newspaper websites. The language of these comments is a mixture of standard Arabic and Colloquial Egyptian Arabic (CEA). The total of 1499 user-generated comments was annotated by a single annotator, a linguist, who was asked to provide one of two tags: insult, not_insult. The annotator was asked to base his tag on the final message of the comment, and make sure back-handed compliments were annotated as insults. We then examined a variety of pre-processing schemes for this variety of Arabic for which no tools seem to be publicly available. We tried orthographic normalization, linguistically-aware stemming, tf-idf, and the presence of any of the members of a pre-defined set of insulting words in the comments. For our experiments, we used the stochastic gradient descent classifier in the Scikit-learn machine learning toolkit [6]. The best overall results, in terms of the f-score, were achieved by the use of loss = hinge (SVM), number of iterations = 10, alpha = 1e-06, penalty = elasticnet, tf-idf, and with within-word character ngrams of 1 to 5. This yielded precision, recall and F-score of 0.85. It turned out that orthographic normalization and stemming did not help achieve overall better results than the simple character ngram method, although each pre-processing methods had its own advantages. The rest of the paper goes as follows: section II describes the data, section III is a brief description of the linguistics of Arabic online comments, section IV introduces the experiments, section V presents the results, section VI is a discussion of the findings, section VII outlines previous studies and section VIII is the conclusion and future research.

II. DATA

The data for this study comprises 1499 user comments (600 neutral/good comments and 899 insulting comments). The comments were collected and annotated by a linguist who was instructed to base his judgment on the final message of the comment, an not just on the lexical units. A comment is derogatory if it insults a person or a group explicitly or implicitly, whether it uses foul language or not. The average number of words per comment is 44.1 with a standard deviation of 54 (minimum = 2, maximum = 607). There is however a noticeable difference between good and bad comments as the average number of words per good comment is 61.7 (std = 70.48, min = 2, max = 607). The average in bad comments is 32.32 (std = 34.86, min = 2, max = 324). This means that good comments are more detailed and reasoned than bad comments.

An example of these insulting comments is:

دول مش بنات دول عاهرات ولابسين بس غطاء للراس مش اكتر ودول مش شباب دول خرفان

(**Buckwalter**: *dwl m$ bnAt dwl EahrAt wlAbsyn bs gTA' llrAs m$ Aktr wdql m$ $bAb dwl xrfAn)*

(Eng: These are not young women, these are prostitutes who happen to wear headscarfs, and those are not young men. These are sheep).

But many insulting comments are not as obvious. For example, in a reply to a suggestion by commentator A, which commentator B did not find convincing, B wrote:

مش معقول يكون تاثير عصير البرسيم قوي المفعول عليكي كدا

(Buckwalter: m$ mEqwl ykwn Esyr Albrsym qwy AlmfEl Elyky kdA)

(*Eng. I did not know clover juice could have such a strong effect on you.*).

Clover is not used as a juice in Egypt, and it is the staple food for donkeys, which are viewed in Egypt as stupid. In fact, "stupid" and "donkey" are near-synonyms in Egypt. It takes quite a bit of inferencing, even for a human, to determine the final message of such a comment.

List of Insults

Arabic is a morphologically complex language, and word lists may not be useful due to the wide range of morphological variation. To overcome this, we construct the word list in a rather different way: (a) We collected a list of headwords, each of which is a stem in the singular masculine form, (b) for each headword, we generated all the possible forms of the words using simple rules that added prefixes and suffixes to form the various gender, number and definiteness forms as well as the possessive and conjunction forms. The process does not take mor-

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 12, June 2014

phological constraints into account. For example, it would produce a noun that has both the definite prefix and a possessive suffix, which is not a possible form in Arabic. In this step, the broken plural form is used as a unique form, (c) The resulting words are checked against a large corpus of Arabic that contains both standard and colloquial varieties, and only those words that occur in the corpus are maintained, (d) the word list is re-touched for manual correction, and the broken plural is mapped to the singular form. The process resulted in 124 head-words, and 1749 forms. An example is the stem ***SayE*** (Eng: vagabond, immoral), for which we have found 13 possible forms in the corpus:

صايع للصايعين والصايع والصايعين الصايع الصايعين صايعه صايعين الصايعه بالصايع وللصايع صايع للصايع

Buckwalter: SayE, llSAyEyn, wAlSAyE, wAlSAyEyn, AlSAyE, AlSAyEyn, ….)

III. LINGUISTICS OF ONLINE ARABIC

While Arabic newspapers use Modern Standard Arabic in their reports, user comments tend to have different characteristics:

(1) The language is a continuum of MSA and colloquials, a perfect example of Arabic diglossia. Since colloquial Arabic has different morphological and syntactic patterns than MSA, using NLP tools, which are mainly intended for MSA usually, produces bad results.

(2) There is orthographic inconsistency in the use of some Arabic letters, and there exist some confusion groups. For example, the group (ا, أ, إ, آ) (Buck: A, >, <, |) tend to be confused for one another. These are usually syntactically and morphologically different. Another group is the (ه,ة) (Buck: h, p) pair. While ***h*** is a pronoun an ***p*** is a feminine marker, these are usually confused in newspaper comments. Newspaper reports, on the other hand, are more standard.

(3) User comments exhibit the usual problems of run-on words, incorrect punctuation or the lack thereof, incomplete sentences, and vowel repetition for emphasis effects.

IV- EXPERIMENTS AND RESULTS

We run a number of experiments to test which features and learner settings yield the best results in discriminating between insulting and non-insulting comments as follows:

-***The baseline classifier***: this is a key word classifier that determines that a comment is insulting if it contains an insulting word; otherwise, the comment is good. The classifier is meant only as a benchmark since insulting comments may contain no insulting words and the reverse is true.

-***Machine Learning Classifiers***: Beyond the baseline classifier, we use Scikit-learn and its SGDClassifier, which implements linear classifiers (SVM's or Logistic regression) with stochastic gradient descent training. To examine what features produce the best results, we use Scikit-learn's grid search to decide between the following feature options: (a) SVM's versus logistic regression, (b) whether to use tf-idf weighting or simple binary features, (c) whether to use words or characters for classification, and (d) the range of possible ngram (between 1 and 5).

(a) ***Basic Classifier***: The basic classifier is a pipeline of pre-processing and cleaning. We use features of words, where a word is a white-space delimited unit, word n-grams, characters, character n-grams, and weighting by term frequency inverse document frequency (tf-idf).

(b) ***Stemmed Classifier***. This is exactly like the basic classier except that we use linguistically meaningful stems rather than words or characters in the classification. There does not seem to be a publicly available stemmer that can handle the language of online comments, which are usually a mix of standard and regional, here Egyptian, colloquial Arabic. There is, however, an Egyptian Arabic morphological segmenter by [5] , which we modify to produce stems rather than segments. In the absence of grammatical (part of speech) categories in the morphological segmenter, we use a simple algorithm by which we choose the longest segment to be the stem. For example, for the word (**ميبكتبش**) (Buck: ***mbyktb$)*** (Eng. He does not write), the segmenter produces ***m+b+y+ktb+$***, and we thus correctly choose ***ktb*** as the stem.

The segmenter is based on the Timbl memory-based learner [3] in a per-letter classification approach in which the character and its preceding five characters and following five characters are used as features. Using the per letter approach, [5] report a word accuracy of 91.9%, where a word is considered correct if all its letters are correctly classified, and a character accuracy of 97.8%.

Although we have no means of evaluating this stemmer, we manually examined 14 randomly chosen comments, and we found that the stemmer produced 17 errors in 434 words, with an accuracy of 96.08%, which means that the longest segment assumption is a valid one. This also means that the stemmer is usable. We have, however, found cases in which the assumption was not valid, such as in the cases where the stem is two letters or less. Although we have not done so, this can be mitigated by providing the stemmer with a list of affixes.

(c) ***Adding Bad Words***: We then combine the best classifier from above (assuming that its is not # 1) with the bad_words list as an extra feature. We assume that this extra information will lead to better results.

(d) ***Orthographic normalization:*** We also run an experiment, with the best settings from above, using orthographic normalization. In this experiment, we add a preprocessing step in which we conflate confusible letters in one form only. In orthographic normalization any of (A, >, <, |) becomes A, any of (p, h) becomes h, and any of {y, Y} becomes Y.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 12, June 2014

V- RESULTS

The baseline classifier seems to yield reasonably good results with an overall f-score of 0.81. The precision is especially high on classifying bad comments (0.93), which means that almost every comment that contains a bad word is a bad comment, although not all bad comments contain bad words as evidenced by the low recall of 0.62. The same is also true for the bad class, albeit with opposite numbers.

Class	precision	recall	f1-score
good	0.62	0.93	0.74
bad	0.93	0.62	0.74
avg / total	0.81	0.74	0.74

Table 1: Baseline Classifier: Only key words are used to decide whether a comment is insulting.

For all the learning experiments above, we use the Scikit-learn machine learning toolkit. We have found that we obtained the best results, in a non-exhaustive grid search, using the SGD Classifier tuned on the training set of one the 10 folds of cross validation, by using loss = hinge (SVM), number of iterations = 10, alpha = 1e-06, penalty = elasticnet, tf-idf, with within-word character ngrams of 1 to 5. In terms of orthographic pre-processing, we found that orthographic normalization did not help the classification.

Class	precision	recall	f1-score
good	0.819	0.791	0.804
bad	0.866	0.884	0.875
avg / total	0.847	0.847	0.847

Table 2: Basic Classifier: Best Results. Word-internal character ngrams of lengths 1 to 5 without orthographic normalization.

Class	precision	recall	f1-score
good	0.729	0.903	0.805
bad	0.931	0.770	0.841
avg / total	0.849	0.826	0.825

Table 3: Orthographic normalization: (p,h) mapped to h, (A,<,>,|) mapped to A, and (Y,y) mapped to y.

Class	precision	recall	f1-score
good	0.792	0.808	0.80
bad	0.871	0.858	0.863
avg / total	0.841	0.838	0.838

Table 4: Stemmed

VI. DISCUSSION

We notice that the baseline classifier has the highest precision in detecting insulting comments, although its recall suffers badly. This means key words alone can be very useful in the classification. One can assume that with better coverage, the precision and recall can even be higher. Although orthographic normalization does not help the f-score in general, it helps improve the precision significantly on the insulting class. We also notice that the precision on detecting the insulting class is always higher than that on the neutral/good class. The evaluation thus differs significantly from one class to another. Based on the results of different settings, it my be tempting to try a combination of these, which we hold for future improvements.

VII. PREVIOUS WORK

There has been quite a few works on Arabic sentiment analysis in general, but we are not aware of any insult detection work apart from the Kaggle competition, which inspired this paper. The Kaggle challenge targeted the English language. The task was described as "The challenge is to detect when a comment from a conversation would be considered insulting to another participant in the conversation. Samples could be drawn from conversation streams like news commenting sites, magazine comments, message boards, blogs, text messages, etc." The Kaggle competition uses the Area Under the receiver operator Curve for evaluation, while we use the precision/recall/f-score metrics as they seem to be more commonly used in computational linguistics research. We have also calculated the AUC for our best scoring experiments. While the best score on English in the Kaggle competition was 0.84, our average AUC is 0.92. The results are, however, not comparable due to the different problems in the two languages.

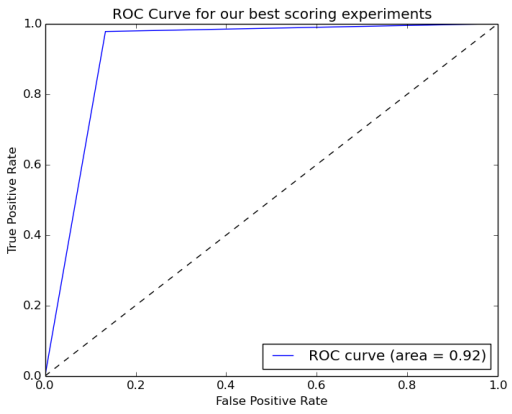

Fig 1: Roc Curve for the Basic Experiment

VIII- CONCLUSION

We have provided a way of automatically detecting insulting comments in Arabic online commentary. We have shown that using stemming and normalization does not

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 12, June 2014

help in general although it boosts the performance on some categories. Simple character ngram models with term frequency inverse document frequency have proved very useful while a list of insulting words did not help. We have also provided a small dataset, and we think a bigger dataset can help improve the performance of the classifier. We plan to improve the data coverage and experiment with other algorithms and settings. A further extension could also include separating the non-insulting category into good and neutral classes.

REFERENCES

[1] Tim Buckwalter (2002). Arabic morphological analyzer version 1.0. Linguistic Data Consortium. LDC Catalogue Number: LDC2002L49.

[2] Walter Daelemans, Jakub Zavrel, Antal van den Bosch and Ko van der Sloot . 2010. TiMBL: Tilburg Memory - Based Learner. Reference Guide.

[3] Kaggle Competition. Detecting Insults in Social Commentary. https://www.kaggle.com/c/detecting-insults-in-social-commentary

[4] Mohamed Maamouri and Ann Bies (2004) . Developing an Arabic Treebank: Methods, Guidelines, Procedures, and Tools, In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004, Geneva, August 28, 2004.

[5] Emad Mohamed, Behrang Mohit and Kemal Oflazer (2012). Annotating and Learning Morphological Segmentation of Egyptian Colloquial Arabic. Proceedings of the Language Resources and Evaluation Conference (LREC), Istanbul

[6] Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research (JMLR). 12, pp. 2825-2830

[7] Princeton University "About WordNet." WordNet. Princeton University. 2010. <http://wordnet.princeton.edu>

AUTHOR'S PROFILE

Emad Mohamed is currently assistant professor of linguistics at Suez University, Suez, Egypt. He obtained a PhD in computational linguistics from Indiana University, Bloomington in 2010 where he wrote a dissertation on Orthographic Enrichment for Arabic. He has since published on word segmentation, diacritization, POS tagging, and dependency parsing for different Arabic varieties. Email: emohamed@umail.iu.edu