Sentence Classification on Hebrew Texts According to Polarity

Tzeviya S. Fuchs[1], Dror Mughaz[1,2]
[1] Dept. of Computer Science, Lev Academic Center, 91160 Jerusalem, Israel
[2] Dept. of Computer Science, Bar-Ilan University, 52900 Ramat-Gan, Israel
tzevia@jct.ac.il, myghaz@gmail.com

Abstract
Research on Sentiment Analysis (SA) has been previously conducted mainly on Latin languages, and none has been done on Hebrew. Additionally SA is usually performed on the document level, rather than on shorter text segments.
In this work, we focus specifically on classifying modern Hebrew short texts according to their polarity. We compare various ML algorithms and techniques of classification, adjust them for Hebrew, propose additional methods, and analyze the differences between our results and others'. In our model, we achieved accuracies higher than those achieved previously in this field.

Background
SA deals with classifying written texts according to their polarity, i.e. as being positive or negative.
Automatic sentiment classification is important for marketing research; it is useful for companies that wish to find out what the world thinks of their products.
Previous research on SA usually focused on classifying long texts written in Latin languages. The approach for short text segments and Semitic languages is different.
Regarding the ML algorithms used for classification, Joachims (1998) identified the benefits of SVMs for text categorization; indeed, it often achieves the best performances.
Yu et al. (2003) achieved 90% accuracy with SA on sentences, in very specific circumstances, using an unsupervised learning method.
Khoo et al. (2006) achieved 88% accuracy in topic-based categorization, using ML supervised algorithms.
In general, SA research shows clearly that one of the most important steps of text classification is correctly extracting the features for the feature vector (i.e., using lemmas, negation tags, etc.). Other experiments involve choosing methods for feature selection, and comparing different ML algorithms.

Objective
In this work, we focus specifically on classifying Modern Hebrew short texts according to their polarity. We test several representation options of the feature vector, as the task seems to be very language-specific. Additionally, we try some new optimizations.
We elaborate on the differences in classifying short texts versus long ones and about the uniqueness of working specifically with Hebrew; that is, how the morphology and even culture of Hebrew writers influence the results of our classification.

Methods
- **Baseline experiment:** We constructed a negative-words list and a positive-words list, composed of the synonyms of the Hebrew words '*good*', '*excellent*', '*bad*' and '*very bad*'. The *decision process* counts the number of positive and negative words that appear in the given texts, and classifies the text as belonging to the class to which most of its words belong.
  The experiment has been performed twice:
  1. With the sentences in their raw form;
  2. After the sentences have been lemmatized.
- **Machine Learning Techniques:** Representation options that have been tested are as follows: tokenization, negation tags, lemmas, binary feature vectors, bigrams, adjectives, stop words removal, and POS tags.

To deal with '*thwarted expectations*', we tried a method that removes the portion of the sentence that appeared before the '*but*'.
Various methods of **Feature Selection** (FS) have been used:
1. FS according to the **rank** assigned to the features by SVM.
2. FS according to the **number** of appearances of a feature in the corpus.

We proposed an additional method:
- FS according to the **position** of a feature in a sentence:
  - Manually identify the object of the sentence;
  - Choose a window size (around the object); words beyond the given range are omitted.

E.g. (windows size = 5, translated):

Features selected — *omitted*

'The <obj> sound quality </obj> is excellent, the best I've heard on mobile devices, especially when you hear music with headphones, because there is an option for Dolby Surround.'

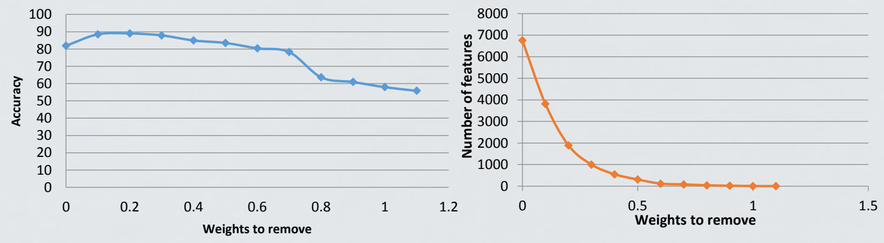
Results
- Accuracies of the *baseline* experiments:

	Classified Correctly	Classified Incorrectly	Unclassified
Raw Form	24.73%	10.49%	64.78%
Lemmatized	34.75%	12.85%	52.40%

- Main results of various vector representations (evaluated using SVM):

	Results of previous research	Our results
Word/ lemma	Lemmas harm classification, using the raw term is better.	Lemmas proved to be effective when used as a form of feature dimension reduction.
Negation tags	Affect classification slightly.	Improved classification, despite inaccuracy of tagging.
FS	Usually doesn't affect when classifying with SVM.	FS by ranks (#1) proved to be extremely affective and increased classification accuracy significantly.

- FS by rank (#1):



As shown in the graphs, nearly half of the features are ranked beneath 0.1; removing them greatly improved accuracy. The reason: the "noise" in the corpus. Previous works state that SVM is not sensitive to FS.
- FS by frequency (#2): removes features needed for classification (slightly harmful due to lengths of texts).
- FS by position (#3): yielded results close to the original ones, and reduced computational load (useful when dealing with long and elaborate sentences).

- While SVMs are considered to be the best classification methods, Bayesian Logistic Regression yielded results almost as high (on average slightly lower).
- The highest results were achieved by Bayesian Logistic Regression, at nearly 93%, when previous somewhat similar research achieved a maximum of 85%-90%.

Conclusions
- The *baseline* experiment's low accuracy rates prove that polarized sentences don't necessarily include positive or negative words.
- *ML methods*' results imply that the feature vectors have much "noise", despite the fact that they represent short text segments.
  - The cause of unnecessary features could be the many inflections and various spelling forms typical for Modern Hebrew.
- Adding negation tags improved classification, despite previous work claiming it is unnecessary. In fact, negation expressions were very dominant among the words of the corpus.
  - This shows the different language structure of Hebrew, and perhaps shows a cultural difference between Hebrew writers and the writers of Latin languages.

Literature Cited
1. Joachims, Thorsten. *Text categorization with support vector machines: Learning with many relevant features*. Springer Berlin Heidelberg, 1998.
2. Yu, Hong, and Vasileios Hatzivassiloglou. "Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences." *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, 2003.
3. Khoo, Anthony, Yuval Marom, and David Albrecht. "Experiments with sentence classification." *Proceedings of the 2006 Australasian language technology workshop*. 2006.