Representations and Architectures in Neural Sentiment Analysis for Morphologically Rich Languages: A Case Study from Modern Hebrew

Adam Amram and **Anat Ben David** and **Reut Tsarfaty**
The Open University of Israel, University Road 1, Ra'anana
adam.amram@gmail.com, {anatbd,reutts}@openu.ac.il

Abstract

This paper empirically studies the effects of representation choices on *neural sentiment analysis* for Modern Hebrew, a *morphologically rich language* (MRL) for which no sentiment analyzer currently exists. We study two dimensions of representational choices: (i) the granularity of the input signal (token-based vs. morpheme-based), and (ii) the level of encoding of vocabulary items (string-based vs. character-based). We hypothesise that for MRLs, languages where multiple meaning-bearing elements may be carried by a single space-delimited token, these choices will have measurable effects on task perfromance, and that these effects may vary for different architectural designs: fully-connected, convolutional or recurrent. Specifically, we hypothesize that morpheme-based representations will have advantages in terms of their generalization capacity and task accuracy, due to their better OOV coverage. To empirically study these effects, we develop a new sentiment analysis benchmark for Hebrew, based on 12K social media comments, and provide two instances thereof: *token-based* and *morpheme-based*. Our experiments show that the effect of representational choices vary with architectural types. While fully-connected and convolutional networks slightly prefer token-based settings, RNNs benefit from a morpheme-based representation, in accord with the hypothesis that explicit morphological information may help generalize. Our endeavor also delivers the first state-of-the-art broad-coverage sentiment analyzer for Hebrew, with over 89% accuracy, alongside an established benchmark to further study the effects of linguistic representation choices on neural networks' task performance.

1 Introduction

Deep learning (Goodfellow et al., 2016) has seen a surge of interest in recent years, transforming all application domains of machine learning. In particular, *neural network* (NN) architectures currently dominate the development of models and applications in NLP, including syntactic parsing (Chen and Manning, 2014; Dyer et al., 2015; Durrett and Klein, 2015), sequence labeling (Grave, 2008), information extraction (Chen et al., 2015; dos Santos et al., 2015), text classification (Lai et al., 2015; Zhang et al., 2015) and sentiment analysis (Dos Santos and Gatti, 2014; Dong et al., 2014).

Much NN work in NLP has been conducted on English, which in turn raises the question whether these methods will be equally effective when applied to languages with different linguistic characteristics than English, and in particular, *Morphologically rich languages* (MRLs) (Tsarfaty et al., 2010). MRLs are languages where word structure holds substantial information, corresponding to multiple meaning-bearing units (morphemes) per space-delimited token. Furthermore, in MRLs word-order may be flexible. These properties may pose challenges to NN models which rely heavily on the *distributional* characteristics of the input tokens. While it is clear that *any* application of NN architectures to NLP may be non-trivial to design, and task performance may crucially depend on non-trivial architectural and modelling choices (Goldberg, 2015), we further ask: should these choices also be affected by the structure and properties of the language?

Proceedings of the 27th International Conference on Computational Linguistics, pages 2242–2252
Santa Fe, New Mexico, USA, August 20-26, 2018.

In this paper we attend to this question by empirically studying the effects of two dimensions of representational and modeling choices on *neural sentiment analysis* for Modern Hebrew — an MRL with interesting word-internal complexities and surface level ambiguity — and for which no sentiment analyzer currently exists.

We study two dimensions of representation choices: the first concerns the representation of the morphologically-rich input signal, *token-based* vs. *morpheme-based*, and the second concerns the level of encoding of the vocabulary items, which could be embedded as complete *strings* or as sequences of *characters*. We study these modeling choices for five architectures (a linear baseline and four NNs: fully-connected, convolutional, recurrent, and bi-directironal recurrent) to observe and analyze emerging trends in performance. In order to empirically evaluate the different models we develop a new and novel benchmark dataset for sentiment analysis in Hebrew, consisting of 12K user-generated comments on official Facebook pages of political figures. The sentiment of each comment has been manually coded by two trained human annotators, and the data themselves have been morphologically analysed and disambiguated, providing us with the opportunity to represent the input at different granularities.

Our experiments show that representation choices affect task performance, and these effects vary with the architectural type. While simple fully-connected and convolutional networks show advantages with the *token-based* representation, RNNs, in contrast, prefer *morpheme-based* settings and token-based bi-RNNs close much of the gap with the morphemes. This is the case for both the *string-based* and *char-based* encoding. We further show that, as in English, CNNs perform particularly well on Hebrew neural sentiment analysis, and conjecture that this is due to the tendency of CNNs to capture relatively large window sequences that are strong predictors, and these windows are, in turn, compatible with token-level granularity. Based on this investigation, we deliver the first sentiment analyser for Hebrew, with over 89% accuracy on analysing sentiment of user-generated comments, defining a new state-of-the-art for Hebrew NLP. A qualitative analysis of a sample of our best model's prediction errors shows that around a half of the remaining errors are also difficult for humans to classify, and require further work on integrating finer-grained aspects of the data.

Our contribution is then manifold. Firstly, we present empirical evidence to the varied effects that morphological information may have on different NN architectures. Secondly, we deliver a neural sentiment analyser for Hebrew, approaching 90% accuracy, on social media content. Finally, we provide a new manually annotated benchmark for Hebrew sentiment analysis, and a strong baseline for further investigation of the *task-representation-architecture* interplay and neural sentiment analysis in particular.

2 Task Description

Sentiment is a subjective attitude towards, for or against, a certain topic. The term *sentiment analysis* refers to the use of natural language processing and machine learning methods for the purpose of identifying or characterizing the sentiment expressed in a given piece of textual data (Pang and Lee, 2008). In this work we focus on sentiment analysis of social media content.

Due to their use by wide audiences, data extracted from social media are valuable sources for studying social, political and economic phenomena. One of the main motivations to analyze sentiment on the social web is *opinion mining*, an approach that challenges the traditional and dominant paradigms in political science, such as opinion polling, surveys or focus groups (Liu, 2012). Affect responses on social media have also been applied to study political behaviour (Ceron et al., 2014), determine organizational legitimacy (Etter et al., 2017), and even predict results of election campaigns (Tumasjan et al., 2010). While Twitter is a social media platform that lends itself naturally to sentiment analysis due to the short, informal character of tweets (Kouloumpis et al., 2011), various sentiment models analysed online conversations on other platforms (Paltoglou and Thelwall, 2012).

Open-source tools and sentiment analysis algorithms are available in several languages, including Arabic (Abdulla et al., 2013; Abdul-Mageed et al., 2014; Al Sallab et al., 2015), but as of yet no sentiment analysis tool is available for Hebrew. This represents a serious gap in the NLP technology available for Hebrew, and moreover, it places a significant barrier on the investigation of political situations through the unfolding conversations in the social web. Here we aim to leverage NN architectures for developing

a sentiment analysis tool for Modern Hebrew, in order to facilitate opinion mining in Israeli social media, and to expand the language technology available for the Hebrew-speaking research community. Since Twitter is not widely used for political discussion in Israel, we develop a new benchmark based on user comments to politicians' pages on Facebook, which is widely used in Israel by politicians and the public.

3 Data Representation

Previous work on *sentiment analysis* in general and on *neural sentiment analysis* in particular focused mainly on English data. Here, we consider Modern Hebrew, a Semitic language with very different characteristics than English. In particular, Hebrew is a *morphologically rich language* (MRL), of which word structure is internally complex and word order is quite flexible. How might these properties affect our modeling decisions when developing a NN-based sentiment analyzer for Hebrew?

Let us begin by theoretically considering the notion of an input token in Hebrew. Because of its rich morphology, any single space-delimited token in Hebrew may contain, in addition to its lexical content, functional clitics (prefixes and suffixes) that correspond to independent stand-alone words in English. To illustrate, the word "*wkftmkti*"[1] may be morphologically segmented into "*w*" (and) "*kf*" (when) "*tmkti*" (supported+1person), and be translated to "and when I supported". Without segmenting the raw tokens into these morphological units, the positive affinity of "*tmkti*" (supported+1person) may be lost on the model, which may have observed "*tmkti*" as a standalone token elsewhere, but not in this composition.

This situation is further complicated by the fact that Hebrew surface tokens are highly ambiguous. Due to the rich morphology and the lack of diacritics in standard textual data, a Hebrew space-delimited token may admit multiple different morphological analyses, only one of which is relevant in context (Tsarfaty, 2006). For example, the token "*frch*" may be a-priory analyzed as the simple verb "*frch*" (infested+3person) or as the sequences "*f*" (that) + "*rch*" (ran+3person+feminine) or "*f*" (that) + "*rch*" (wanted+3person+masculine). The latter analysis has a stronger affinity with positive sentiment than the others. This means that improper morphological disambiguation, or lack thereof, may undermine sentiment accuracy scores.

When constructing a vocabulary and defining the alphabet on which the NN will operate, we need to make representation choices that would be suitable for these properties of MRLs. In this paper we consider two dimensions of representational choices:

- **Input Items:** *Token-Based vs. Morpheme-Based.* First, we compare NN models trained on a signal consisting of the *raw tokens* with ones trained on sequences of *morphological segments* that represent standalone lexical and functional units. The hypothesis is that models trained on morphologically segmented input will provide better generalization capacity for the network and will thus improve the prediction accuracy.

- **Vocabulary Encoding:** *String-Based vs. Char-Based.* It has been proposed in previous work (Ling et al., 2015; Ballesteros et al., 2015) that combating the complexity of word structure in NN architectures may benefit from encoding the vocabulary using sub-words or character sequences. We thus contrast models that encode vocabulary items as complete strings to ones that encode a vocabulary of characters that are used to construct each of the strings.

The empirical investigation we conduct aims to empirically answer two questions: (i) would the choice of representation affect prediction accuracy? and, (ii) would the different representation choices affect different neural network architectures *in different ways*?

4 Data Preparation

In order to empirically evaluate task performance for the different representational choices and architectural design, we develop a new Hebrew benchmark for sentiment analysis and provide these annotated data in two different representational forms: a *token-based* and a *morpheme-based* version.

[1] We assume the transliteration of Sima'an et al. (2001).

Positive Sentiment	Negative Sentiment
"Ruvi, well done! Keep following your predecessor. Talk less and do more. You are the right person in the right timing."	"We should revenge the entire village of the kidnappers. Without fear. A government of cowards."
"Mr. President, you radiate peacefulness, humanism and security, and I hope this will soon be the case. Bless you. Mr. President, your vigorous actions create waves of connection and healing. I thank you for being my president."	"Shame on you! We don't care about your opinion! Constitutionally, the president should be a-political and your entire stupid post is wrapped with righteous politics. Shame on you and may this soon happen to your granddaughters".

Table 1: Inter-Annotator Agreement on the Rivlin's Annotated Corpus

Coder 1, Coder 2	N cases	Example	Explanation
Positive, Negative	26	"Terrible. May he rest in peace."	Mixed sentiment
Negative, Positive	14	"I am dreaming about peace and you are dreaming about football! A wasted dream!"	Potentially sarcastic
Neutral, Negative	2	"Daniel, this is not the time to generalize people despite of what you think of him. He's a Jew just like you and me."	Internal discussion among commenters
Neutral, Positive	2	"Mr. President, how can I arrange an appointment with you? I have been cherishing you for many years, since eleventh grade."	Off-topic
Positive, Neutral	4	"Mr. President, I would like to ask you to help the communities surrounding the Gaza Strip and to provide them with adequate housing and educational services for their children in safe places until genuine calm. This is not a matter of abandonment and Hamas propaganda is baseless."	Off-topic
Negative, Neutral	2	"Why not the length and breadth of it, Mr. President"	Potentially sarcastic

Table 2: Qualitative analysis of a sample of 50 comments for which there was no inter-rater agreement

Our data set consists of 12,804 user comments to posts on the official Facebook page of Israel's president, Mr. Reuven Rivlin. In October 2015, we used the open software application Netvizz (Rieder, 2013) to scrape all the comments to all of the president's posts in the period of June – August 2014, the first three months of Rivlin's presidency.[2] While the president's posts aimed at reconciling tensions and called for tolerance and empathy, the sentiment expressed in the comments to the president's posts was polarized between citizens who warmly thanked the president, and citizens that fiercely critiqued his policy. Of the 12,804 comments, 370 are neutral; 8,512 are positive, 3,922 negative.

We use a morphosyntactic parser called *yap* (*yet another parser*) (More and Tsarfaty, 2016) to morphologically analyze, disambiguate and segment all comments in our dataset. In the token-based representation, there is an average of 23 tokens per comment, and in the morpheme-based representation an average of 31 morphemes per comment. In terms of *out-of-vocabulary* (OOV) items, we observe a higher OOV rate in the token-based test set, 8.865% (2552 out of 28787), compared to the OOV rate 7.624% (1442 out of 18912) with the respective morpheme-based representation of the same set, as expected. So, at least in theory, a morpheme-based representation may better generalize from training instances, at least in cases where the composition of seen morphemes yields unseen tokens.

Data Annotation: A trained researcher examined each comment and determined its sentiment value, where comments with an overall positive sentiment were assigned the value 1, comments with an overall negative sentiment were assigned the value -1, and comments that are off-topic to the post's content were assigned the value 0. We validated the coding scheme by asking a second trained researcher to code the same data. There was substantial agreement between raters (N of agreements: 10623, N of disagreements: 2105, Coehn's Kappa = 0.697, p = 0).

Table 1 shows examples of positive and negative sentiment in clear cases of agreement between human annotators. We further examined the source of disagreements between the annotators. In a qualitative analysis of a sample of 50 comments for which there was disagreement between raters, summarized in

[2]It should be noted that the period of data collection was politically and socially charged. This has been due to a series of violent events that escalated fragile tensions with regards to Jewish-Arab relations in the region, including the kidnapping and murder of three teenagers living in the settlements of the West Bank, a revenge murder of an Arab teenager in East Jerusalem, a two-months war in Gaza, and a controversial wedding between a Jewish woman and an Arab man which sparked demonstrations led by extremist Jewish groups.

Table 2, we observe that the majority of the disagreement cases originated in comments that contained mixed sentiments, such as in the following:

- "Mr. President, every person has the right to decide on their lives, but converting to Islam in these crazy days, which is what her partner wanted, means that their offsprings will not be Jewish. You are talking about racism but they want to destroy us because we are Jews. Mr. President I wish you success, health and joy".

- "The hotheads, the enemy inside. The real enemy 'thanks' to whom we were scattered in all directions in the era of the Temple, the real enemy that might put us all in the grave".

In the first example, the commenter expresses a positive and respectful sentiment towards the president, but displays a negative sentiment towards conversion to Islam. In the second example, the sentiment can be interpreted in both directions. The content of the comment expresses negative sentiment towards the 'hot heads', but reading in also the context on the president's post, it actually expresses positive sentiment towards the idea that tolerance is better than hatred. It appears then that the unrestricted length and form of expression in Facebook comments makes the overall sentiment classification task more challenging.

Data Normalization and Two-Level Representation: We excluded from the dataset comments that do not contain any word in Hebrew. We also added spaces to separate Hebrew (or English) tokens, numbers, and punctuation symbols from one another in the text sequence. We then fed the raw tokens in our dataset to a morphological analysis and disambiguation parser (More and Tsarfaty, 2016), to build the respective morpheme-based representation. We split our dataset into a *train set* (80%) and a *test set* (20%) of sizes 10,244 and 2,560 comments, respectively. Out of the comments in the *train set*, 20% are reserved as *dev set* for evaluation and optimization after each epoch.

5 Neural Network Architectures for Sentiment Analysis

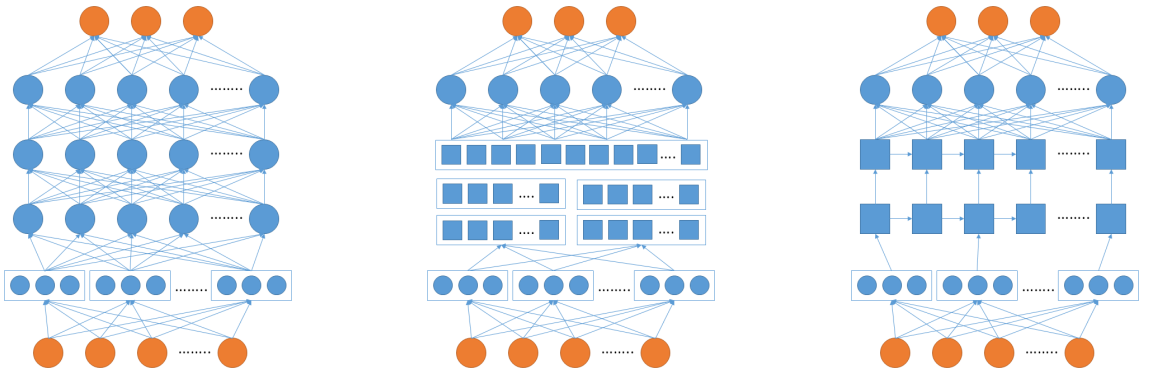
For each of the representation choices outlined above, we set out to compare and contrast the performance of different architectures. In all cases, we learn from our data a sentiment analysis function $f : x \rightarrow y$, where $x \in \mathcal{X}$ is a textual sequence which may be represented and encoded in the different ways discussed in Section 3, and $y \in \{$*positive, neutral, negative*$\}$ is a three-way classification of sentiment polarity.

We compare traditional linear models with non-linear models, contrasting different choices of NN architectures. Simple feed-forward fully-connected networks, such as the *multi-layer perceptron* (MLP), accept a sequence of items as input, which are then treated as features of the model (a *bag of words* (BOW) approach). They are simple to conceptualise and construct, however, the BOW representation, while highly sensitive to lexical content, is insensitive to any sequencing or ordering preferences. So, "The soup was not good, it was bad" may score the same as "The soup was not bad, it was good".

To capture some ordering information, it is customary to employ a feed-forward convolutional neural network (CNN) with a pooling layer. CNN architectures apply a non-linear function on a sliding window over words in the comment, and transform it to channel size vector. Then, a "pooling" operation combines the different vectors into a single channel size vector, that is in turn used for prediction. CNNs were previously shown to provide excellent performance on sentiment analysis of English (dos Santos et al., 2015). Would it be possible to replicate these achievements for Hebrew?

Ordering information may alternatively be captured by changing the network architecture to explicitely encode a sequence: this can be done using *recursive* or *recurrant neural networks* (RNNs). Indeed, various English sentiment analyzers assume RNNs at their backbone (Dong et al., 2014). For sentiment analysis we assume an RNN architecture, where the intermediate states throughout the sequence are used for prediction, loss calculation, and back propagation to update previous states.

In this work we consider a concrete implementation of RNNs — *Long Short-Term Memory* (LSTM) networks (Hochreiter and Schmidhuber, 1997) — which preserves gradients over time using functions that simulate logical gates (memory cells). At each input state, a gate is used to decide how much of the new input should be written to the memory cell, and how much of the current content of the memory



(a) The MLP Architecture (b) The CNN Architecture (c) The RNN Architecture

Figure 1: The Neural Network architectures in our experiments. All architectures started with an input vector of word/char index followed by an embedding layer with embedding size 300. In (a) we used 3 fully-connected layers, the first with 256 units, the second with 128 units and the third with 64 units. Each fully-connected layer has a dropout rate of 0.5. (b) uses 2 parallel convolution layers with 128 filters and kernel size of 3, 8 for *string-based* and 10, 30 for *char-based*. After the convolution, we apply max pooling with pooling size of 2. We then concatenate the results from previous layers and feed it to a fully-connected layer with 128 units and 0.5 dropout rate. (c) uses 2 layers of specific RNN implementations, LSTM or BiLSTM, followed by a fully-connected layer with 128 units and 0.5 dropout.

should be forgotten. Three gates are controlling for input, forget and output. Gate values are computed based on linear combinations of the current input and the previous state.

We also consider *Bidirectional-RNNs*, and in particular, a *Bidirectional-LSTM* (BiLSTM), which is a variant of LSTMs that has been shown to perform well on various sequence labeling and transduction tasks. BiLSTMs maintain two separate states for each input position, one forward and one backward. The forward and backward states are generated by two different LSTMs. The first LSTM is fed the input sequence as is, while the second LSTM is fed the input sequence in reverse. For morphologically rich sequences as discussed here, BiLSTMs may potentially capture the influence of prefixes and suffixes on upcoming content as well content that precedes them. At least in theory, such effects may help leverage morphological phenomena for classification.

6 Experiments

We set out to evaluate the effects of different input representation choices (Sections 3) and different architectural choices (Section 5) on neural sentiment analysis for Modern Hebrew. All of our models aim to learn the same objective function $f : x \rightarrow y$ where $x \in \Sigma^*$ is a sequence of vocabulary items over an alphabet Σ, encoded in the different ways detailed in Section 3. The output is a sentiment value $y \in \{positive, neutral, negative\}$.

We compare a traditional **Linear** basedline model to four different NN-based models: **MLP, CNN, LSTM** and **BiLSTM**. We implemented our models using Keras (Chollet, 2015) a high-level API with TensorFlow (Abadi et al., 2015) running as the backend engine. All models start with input represented by a vector of word/char index which is fed into an embedding layer, followed by the specific model layers (e.g. convolutional, RNN, etc.), and conclude with a softmax activation for the output. All of our NN models' architectures are depicted in Figure 1.

The **Linear** model contains a single fully-connected layer with 100 units and without activation. This is the only model that did not include an embedding layer and dropout regularization. The **MLP** model contains 3 fully-connected feed-forward layers, the first with 256 units, the second with 128 units and the third with 64 units. We apply dropout with rate of 0.5 on each fully-connected layer. Our **CNN** (ConvNet) contains two convolutional layers, one with kernel size 3 and the other with kernel size 8. After each convolutional layer there is a max pooling layer. These layers are followed by a fully-connected layer

(a) *String-based Vocabulary*					
Architecture: *Input Representation:*	**Linear**	**MLP**	**CNN**	**LSTM**	**BiLSTM**
Token-Based	68.20	**86.80**	**89.20**	85.20	87.40
Morpheme-Based	66.13	86.40	89.06	**86.20**	**87.50**
(b) *Char-based Vocabulary*					
Architecture: *Input Representation:*	**Linear**	**MLP**	**CNN**	**LSTM**	**BiLSTM**
Token-Based	**69.38**	74.60	82.40	69.50	73.70
Morpheme-Based	68.71	74.50	78.55	70.60	73.10

Table 3: Accuracy results (percentage of correct label predictions) for all architecture and representation choices on our test set; for (a) the *string-based* vocabulary, and (b) the *char-based* vocabulary (b).

with 256 units. For *char-based* we changed the kernel sizes to 10 and 30.[3] Our **RNN/BiRNN** Recurrent Neural Network architectures contain two specific RNN implementations (LSTM or BiLSTM) with 93 units followed by 1 fully connected layer with 256 units and 0.5 dropout rate.

All models use the cross-entropy loss for training and Adam optimizer (Kingma and Ba, 2014) to update models weights and biases. We set a learning rate of $1e-4$, and use batch size of 50 for training.

We evaluate two alternatives for representing the input: a *token-based* representation, where raw tokens are passed on as they are, and a *morpheme-based* representation, where the input signal is first morphologically disambiguated and is passed on as a sequence of morphological segments. From the train dataset we created a vocabulary Σ with 5K top items (tokens/morphemes). Then, we considered two alternatives of encoding the vocabulary items. A *string-based* encoding, where each of the tokens/morphemes is represented as a single item in the vocabulary, and a *character-based* representation assuming a narrower vocabulary, consisting of alphanumeric characters and special signals, and where each item (token or morpheme) is represented as a sequence of characters. The character set used in the vocabulary of our char-based models consists of 220 characters, including Hebrew letters, digits, and other characters such as white space and a new line character.

We set the comment length limit to 100 items (tokens or morphemes) in the *string-based* encoding and to 300 for *char-based* encoding. If a sequence was less than the limit, we padded it with the padding value, and if it is greater we trimmed it.

Results: Table 3(a) presents the results of the empirical comparison between our five architectures, in *token-based* vs. *morpheme-based* settings, for the *string-based* encoding of vocabulary items. When contrasting *token-based* and *morpheme-based* NN architectures in *string-based encoding*, the most striking outcome is that representation choices do have an effect on task performance, and that the difference in task performance varies across architectures.

First of all, we see that linear models, assuming a naïve classification architecture based on unigram features, perform roughly at the same level as a "majority vote" baseline, i.e., a classifier that would simply predict "positive" for all cases. In our data, "positive" constitutes 66.6% of the gold judgments. Yet, we observe a clear empirical advantage to the *token-based* representation over *morpheme-based* in this setting, perhaps since tokens allow more context to enter the classification — these tokens may essentially be seen as n-grams of morphological segments. Furthermore, in feed-forward networks, both fully-connected and convolutional, there is still an advantage to the token-based representation. We conjecture that this is due to the inherent order-insensitivity of these models. Morphological segments out of context may be less informative, and tokens capture more context of particular morphemes, where these larger "chunks" may exhibit particular ngrams that are strong predictors.

RNNs, in contrast, prefer the morpheme-based setting, and in the LSTM this is in a non-negligible margin. This is in line with our hypothesis that explicit modeling of morphology may capture interactions of elements in the sequence and allow for better generalization thereof. In contrast with MLPs and CNNs, RNNs consider the entire sequence for classification, and so it may be the case that previously unseen tokens in the sequence may undermine classification in token-based settings. This RNN representation

[3]This design outperformed CNNs with smaller (3,4,5) and larger (5,10,15) kernel sizes in our preliminary experiments.

can benefit from the more compact vocabulary of the morpheme-based models, and may also benefit from the decomposition of unknown tokens into potentially known morphemes to better generalize from seen sequences to unseen ones. That said, it is to note that token-based BiLSTM closes much of the gap with the morpheme-based setting, which suggests that the bidirectional representation of tokens perhaps already implicitly captures some aspects of tokens' internal decomposition and morphological signature.

The best result across all models is obtained with the token-based CNN. As in English, CNNs show excellent performance for this task, providing state of the art results for Hebrew sentiment analysis, above 89% accuracy. In particular, this is obtained in the token-based settings, where tokens are compatible with the "n-gram" views filtered through the convolution.

Table 3(b) presents the results of the same comparisons, for *char-based* vocabulary encoding. Interestingly, for the linear model the char-based representation outperforms the string-based counterpart, but these models still lag far behind the rest of the neural networks. With simple and convolutional feed-forward networks we continue to see that token-based representations outperform. Yet, in all cases, we see better scores for the parallel models in the *string-based* setting. RNNs are inferior to all CNNs and MLPs, still presenting an advantage to the morpheme-based representation, and almost no difference in the BiLSTM experiment. It seems that while characters may be able to capture some (morphological) regularities inside tokens, they are not very good at capturing the content of long character sequences.

For our best model overall, a token-based CNN in the string-based lexicon encoding, we compared the automatic prediction on a sample of 50 examples with the manually annotated sentiment. 26/50 (52%) of the mis-classifications are in cases when the human raters also happen to disagree, mostly due to mixed sentiment. This suggests that further improvement will require more sophisticated, aspect-based modeling, on finer-grained sentiment targets.

The main message coming out of this investigation is that the representation of linguistic items (tokens, words, morphemes) may influence the performance of NN-based architectures in the case of MRLs, and that the extent of the difference in performance depends on the type of architecture. It should be noted though that this outcome is — quite possibly — task-dependent. Sentiment analysis presents a three-way classification over complete sequences, for which long ngrams often seem to serve as good predictors. It may well be that for sequence transduction (as in SMT) or structure prediction (as in parsing) we will observe different trends in the influence of representation types on different languages and architectures.

Error Analysis: To gain further insight into the difference in task performance between different models and architectures, we performed a qualitative error analysis for 6 of our best models: the **MLP**, **CNN**, **LSTM** architecture, in *token-based* and *morpheme-based* settings, for the *string-based* lexicon.[4]

In general, all models are better at identifying positive sentiment than in identifying negative sentiment. The cases where both morpheme-based and token-based models correctly identified the *positive* sentiment constitute 91% of the *positive* comments in the MLP case, 93.8% in the CNN, and 94.1% in the LSTM. In contrast, morpheme-based and token-based models correctly identified *negative* sentiment only 27.8% of the *negative* comments in the MLP, 30.1% in CNN, and 48.1% in the LSTM. Cases where morpheme-based and token-based models wrongly identified a positive sentiment as negative one have different characteristics based on the architecture. In the MLP these are characterized by a double negation, and by addressing a third party. For CNN, comments that start with "Sorry, but.." are identified as negative, even when these happen to be positive. For the LSTMs, we see error in the prediction of positive sentiment in the case of extremely long comments, i.e., accuracy here is length sensitive.

We now turn to consider cases where the token-based models made the right prediction and morpheme-based models made the opposite prediction. The morpheme-based MLP wrongly identified a positive sentiment as a negative one in the case of short comments, many punctuations, and many negation elements. The CNN wrong classification of this type is characterized by a double negation, and also by addressing multiple parties in the same comment. For the RNN, this wrong prediction is again characterized by length — at moderate length of 12 morphemes the model often doesn't succeed in making the correct prediction. The opposite case, where morpheme-based models identify negative sentiment as a positive one has a different characterization in these architectures: in the MLP this is characterized

[4]For the complete report, confusion matrix, error characterization and translated examples, see our supplementary materials.

	MLP	CNN	LSTM
MB,TB-correct,pos	60.39 %	62.07%	62.4%
MB,TB-correct,neg	23.25%	24.14%	14.08%
MB,TB-should be pos	3.59% double negation	2.34% "sorry, but.."	2.12% very long comments
MB,TB-should be neg	4.2% honorary, criticism, !!!	3.125% sarcasm, irony	7.185% address+mixed modifiers
TB-pos MB-error,neg	1.09% short, honoraries+negations	0.39% double negation, third party	1.4% short-moderate length
TB-neg MB-error,pos	1.95% honorary title+negative words	2.3% mixed sentiments	0.93% honorary title+opposing words
MB-pos TB-error,neg	1.25% third party, neg words	1.52% direct address / "advise"	1.09 sequential short utterances
MB-neg TB-error,pos	1.36% third party ref	1.093% repetitions, moderate length	7.92% very short comments

Table 4: Qualitative Error Analysis and Confusion Quantitative Assessment on 2560 comments. TB denotes *token-based* and MB denote *morpheme-based* representation. We do not discuss neutral sentiment.

by kind and respectful words towards the president, alongside expressed criticism. For the CNN we see confused classification in comments that express mixed sentiments towards different topics. For the LSTM, we see the same pattern as the MLP; respectful attitude towards the president (Dear Mr. president, respected president) followed by an opinion which is opposed to the post's actual content.

Finally, cases where morpheme-based models made the right prediction and token-based models were erroneous, i.e., the cases that coincide with our linguistic hypothesis, can be characterized as follows. For the MLP, positive sentiments where identified as negative where the comments address a third party and also contain certain negation elements (i.e., "I choose Rivlin despite what Liberman says"). CNNs classified positive sentiments as negative where the comments directly address the president, proposing a sort of explicit "advice" ("Dear Mr. president we need to stop the violance"). The LSTM gets confused in cases of long comments consisting a sequence of very short utterances, typically also containing many negations and punctuation ("Me too. I'm opposed. Where do we go?" etc.) The opposite error also happens in the token-based models, where morpheme-based are correct. MLPs classify negative as positive where the comments do not address the president, but speak of a third party (the children, the soldiers, etc). Token-based CNNs mistakenly classify negative as positive in comments of moderate length (about 14 words) or those that have many repetitions ("sit down sit down sit down quietly and listen.."). The token-based LSTM is wrong in cases of *very* short comments, with only a handful of tokens.

All in all morpheme-based models do show advantages in identifying the correct sentiment of actual content (addressing of third parties, serving advise, etc.) beyond fixed expressions, and are better in classifying short texts or texts with repetitions. However it seems that the morpheme-based models do get more easily confused by the existence of negation elements, double negation, and many titles honoring the speaker (the president). These may steer the classification in a fairly rigid direction. Token-based models that consider larger n-grams, in particular in the context of CNN architectures, capture larger context windows, which currently present the best performance on Hebrew sentiment analysis.[5]

7 Discussion and Conclusion

Despite the great success of NN-based methods in NLP, the question of the interplay between *language type* and *architectural choices* has only been scarcely attended to, if at all. Yin et al. (2017), for instance, compare the strengths of different NN architectures for NLP tasks, but the discussion focuses solely on English. Dos Santos and Zadrozny (2014) propose char-based CNNs for NLP, but the benefits of the char-based modeling are not contrasted for different NN architectures and modeling. Finally, Al Sallab et al. (2015) contrast deep learning architectures for sentiment analysis in Arabic. They use a simple *bag of words* (BOW) approach without considering the MRL nature of the language, nor they are comparing different representation choices for their lexicon.

In this work we address neural sentiment analysis of Hebrew, a Semitic language known for its morphological complexities, and evaluate two choices of representing the input signal: (i) tokens vs. morphemes input, and (ii) string-based vs. char-based encoding. We experiment with different architectures: a linear model, feed-forward MLPs, CNNs, and (Bi)RNNs. We show that while linear models as

[5]Our data set, code, and models are publicly available at `https://github.com/omilab/Neural-Sentiment-Analyzer-for-Modern-Hebrew`.

well as MLPs and CNNs obtain higher accuracy at token-level granularity, RNNs prefer the morpheme-based settings. We further show that while char-based representation can in most cases dispense with morpheme-level information, it comes at a cost of an overall drop in accuracy. Our best model is a token-based CNN with above 89% on a new Hebrew benchmark based on social media content.

We believe that this investigation is only a first step in un-black-boxing the use of NN models for language processing tasks. Through this investigation, the two-level sentiment benchmark we deliver, and the strong baseline results we provide, we hope to encourage further investigation into the interplay between *task, language types* and *modeling choices* in general, and on NN models for MRLs in particular.

Acknowledgments

We thank Tzipy Lazar-Shoef for research assistance, and are thankful to three anonymous reviewers for their insightful comments. This research is funded by the Israel Science Foundation, ISF grant 1739/26, for which we are grateful.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

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

N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6, Dec.

Ahmad A Al Sallab, Ramy Baly, Gilbert Badaro, Hazem Hajj, Wassim El Hajj, and Khaled B Shaban. 2015. Deep learning models for sentiment analysis in arabic. In *ANLP Workshop 2015*, page 9.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. *CoRR*.

Andrea Ceron, Luigi Curini, Stefano M Iacus, and Giuseppe Porro. 2014. Every tweet counts? how sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to italy and france. *New Media & Society*, 16(2):340–358.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*, pages 167–176. The Association for Computer Linguistics.

François Chollet. 2015. Keras. `https://github.com/fchollet/keras`.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland, June. Association for Computational Linguistics.

Cícero Nogueira Dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78.

Cícero Nogueira Dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1818–II–1826. JMLR.org.

Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. *CoRR*, abs/1504.06580.

Greg Durrett and Dan Klein. 2015. Neural CRF parsing. *CoRR*, abs/1507.03641.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *CoRR*, abs/1505.08075.

Michael Etter, Elanor Colleoni, Laura Illia, Katia Meggiorin, and Antonino D'Eugenio. 2017. Measuring organizational legitimacy in social media: Assessing citizens' judgments with sentiment analysis. *Business & Society*, page 0007650316683926.

Yoav Goldberg. 2015. A primer on neural network models for natural language processing. *CoRR*, abs/1510.00726.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep learning. Book in preparation for MIT Press.

Alex Grave. 2008. *Supervised Sequence Labelling with Recurrent Neural Networks*. Ph.D. thesis.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11(538-541):164.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2267–2273. AAAI Press.

Wang Ling, Tiago Luís, Luís Marujo, Ramón Fernández Astudillo, Silvio Amir, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Finding function in form: Compositional character models for open vocabulary word representation. *CoRR*, abs/1508.02096.

Bing Liu. 2012. Sentiment analysis and opinion mining.

Amir More and Reut Tsarfaty. 2016. Data-driven morphological analysis and disambiguation for morphologically rich languages and universal dependencies. In *Proceedings of COLING 2016*, december.

Georgios Paltoglou and Mike Thelwall. 2012. Twitter, myspace, digg: Unsupervised sentiment analysis in social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):66.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January.

Bernhard Rieder. 2013. Studying facebook via data extraction: the netvizz application. In *Proceedings of the 5th annual ACM web science conference*, pages 346–355. ACM.

Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. 2001. Building a tree-bank of modern hebrew text.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Yannick Versley, Marie Candito, Jennifer Foster, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (spmrl) what, how and whither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 1–12, Los Angeles, CA, USA, June. Association for Computational Linguistics.

Reut Tsarfaty. 2006. The interplay of syntax and morphology in building parsing models for modern hebrew. In *Proceedings of ESSLLI*.

Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10(1):178–185.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of CNN and RNN for natural language processing. *CoRR*, abs/1702.01923.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.