Semitic Root extraction as a Classification Task

Lena Rampula

Introduction

- Following the work of Daya et al. *(2008)*:

  Identifying Semitic Roots: Machine Learning with Linguistic Constraints.
- A different take on the same problem, using a different data set and a different learning environment.
- Work in progress.

Introduction

- Root extraction is an important task when dealing with Semitic languages.

  - Roots hold an abstract meaning component.

  - Frequently, words with similar root are semantically related, sometimes in a metaphorical sense
- Previous works on root extraction are dependent on large-scale lexicons
- This work presents a machine learning approach in order to identify the roots of Semitic words.

Why this is important?

- It was shown that same root words facilitate word recognition, while phonetic, semantic or form similarity do not (Frosts, 2000).
- Berman (2003) showed that children acquiring Hebrew use root and pattern knowledge for word formation.
- An important contribution to IR and other computational tasks.

Hebrew Morphology

- As in English Hebrew words can have prefixes and suffixes, usually for inflection.
- A lot of the stems in Hebrew can be broken down to a root and a pattern (but not all).
- The root consists of consonants only, by default three
- The pattern is a combination of vowels and consonants, with non-consecutive “slots” into which the root consonants are inserted.

Hebrew Morphology

h.š.v – root

m a _ _ e _

m a h š e v

computer

_ o _ e _

h o š e v

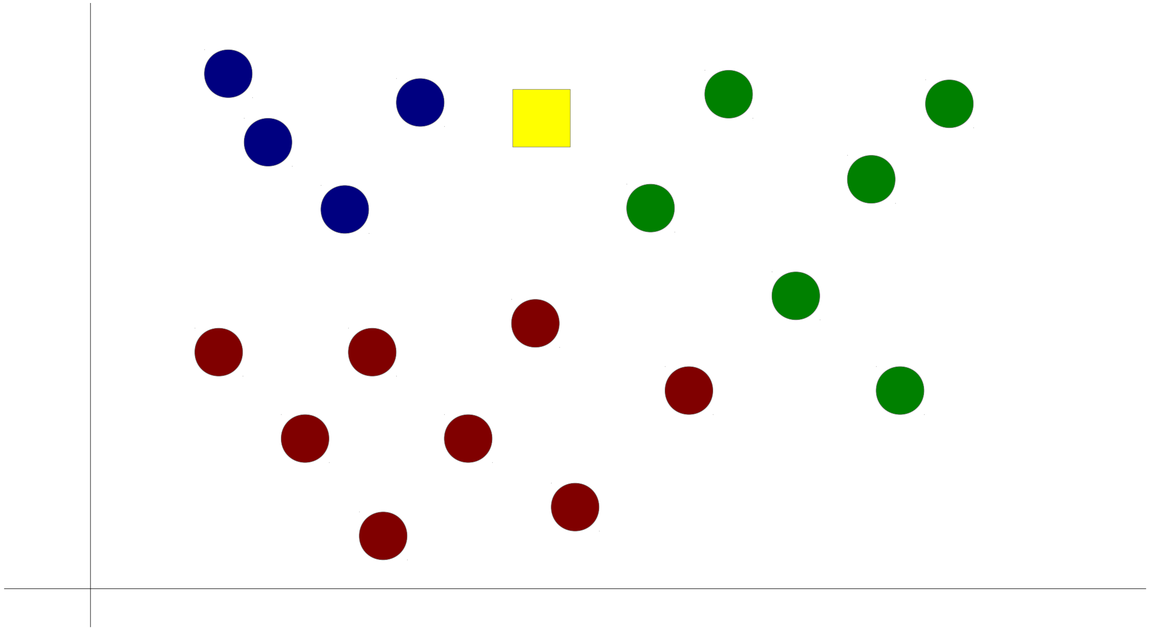
think

Orthographic representation of Hebrew

- The orthographic representation is far too complex to discuss here.
- The complexity leads to multiple possible readings – each reading implies a different root.
- In this work I used a transcribed text, avoiding most of the ambiguous cases

What does a classifier do?

- For simplicity let's look at an intuitive example:



We have 3 classes – blue, red and green

and two features, one on the x and on the y.

What does a classifier do?

- Given a set of classes, we need to determine which class a new case (the square) belongs to.
- The classifier learns the probability that the new case belongs to either red, blue or green, given the features.
- What it has to do with root extraction?

Root extraction as a classification task

- The roots are our classes.
- We need an annotated data set.
- And decide which features are going to help us identify the correct root.
- Now all the classifier needs to do is to build a model based on the seen data, so new cases could be classified.
- How to build such a model?

Naive Bayes Classifier

- A simple probabilistic algorithm based on applying Bayes' theorem with strong independence assumptions.
- It assumes that the value of a particular feature is unrelated to the presence or absence of any other feature, given the class

Advantages

- Requires a small amount of training data.
- Naive Bayes can be applied to many different learning problems, and is unlikely to produce completely failing classifiers
- It can handle a lot of noise in input data.
- Deals well with a lot of features that can have a lot of values.

Naive Bayes Classifier

- If you were wondering where are the statistics, here they come...
- We want to calculate the probability of a new case being in a class *c* given the feature values:

  $p(C|F_1,....,F_n)$

- It is infeasible to calculate this probability when we have a lot of features, with multiple values.

Naive Bayes Classifier

- The Bayes Theorem can help us here:

$$p(C|F) = \frac{p(C)(p(F)|p(C))}{p(F)}$$

$$posterior = \frac{prior * likelihood}{evidence}$$

- And for the multiple features:

$$p(C|F_1, ..., F_n) = \frac{p(C)\, p(F_1, ..., F_n|C)}{p(F_1, ..., F_n)}$$

Naive Bayes Classifier

- Fortunately we do need to calculate the denominator since it is constant, and does not depend on the class.
- But, how to calculate the upper part of the equation?
- The Chain Rule

The chain Rule

- We can rewrite the nominator:

$$p(C)\,p(F_1,\ldots,F_n|C) = p(C)\,p(F_1|C)\,p(F_2,\ldots,F\,n|C,F_1)$$

¿ $$p(C)\,p(F_1|C)\,p(F_2|C,F_1)\,p(F_3,\ldots,F\,n|C,F_1,F_2)$$

¿ $$p(C)\,p(F_1|C)\,p(F_2|C,F_1)\ldots p(F_n|C,F_1,F_2,F_3,\ldots,F_{n-1})$$

- This still looks very hard, how did it help us?

Back to Bayes

- Remember that Naive Bayes is naive? Assuming that the features are independent of each other:

$$p(C)\,p(F_1,\ldots,F_n|C)\,\alpha\,p(C)\,p(F_1|C)\,p(F_2|C)\ldots$$

$$\alpha\,p(C)\prod p(F_i|C)$$

- This means to multiply the prior probability of the class by all the probabilities of a feature given that class.

In Other Words

- Each probability $p(F_i|C)$ – the likelihood - is a weight that indicates how good an indicator F_i is for class C.
- Similarly, the prior p(C) is a weight that indicates the relative frequency of class C.
- More frequent classes are more likely to be the correct class than infrequent classes.
- The product of the prior and the likelihood is a measure of how much evidence there is for the new case being in the class.

Back to the Roots

- The prior probability:

$$\frac{number\ of\ occurrences\ of\ a\ root}{number\ of\ roots}$$

- The likelihood probability:

$$\frac{number\ of\ occurrences\ of\ a\ feature\ with\ a\ root}{number\ of\ occurrences\ of\ a\ feature\ with\ all\ the\ roots}$$

- For each word, we calculate the probability that it belongs to each of the root classes, and choose the root with the highest probability

Data

CHILDES Corpus – Hebrew MOR annotated transcribes (MacWhinney, 2000 ; Berman, 1990).

- Longitudinal study of four children (1;4 – 3;3)
- Child directed speech
- Extracted a list of words containing a root and the context words around it.
- For this demonstration only a partial data set was used.

features

- Location of characters
- Prefixes and suffixes
- Context words
- POS of the word
- POS of the context words
- Still working on more interesting features

Evaluation

- Building the classification model and testing it on the same data is a methodological mistake.
- The model would just repeat the classes of the data that it has just seen and but would fail to predict anything on unseen data.
- To avoid it we divide the data into training data set and test set.
- But, if you run a lot of tries and test each time on the test set, you over-fit the test set.
- The test set should be tested only once.

Cross Validation

- The training set is split into k sets.
- A model is trained using k-1 of the sets as training data.
- And the remaining set is used for testing.
- This is repeated for all the k sets.
- Each round is reporting the error of classification as the classifier performance measure.
- Cross-validation measure is then the average of the values computed for each set.

Full Feature Set Results

- === Stratified cross-validation ===
- === Summary ===
-
- Correctly Classified Instances 7341 79.4911 %
- Incorrectly Classified Instances 1894 20.5089 %
- Kappa statistic 0.7905
- Mean absolute error 0.0009
- Root mean squared error 0.0276
- Relative absolute error 21.1611 %
- Root relative squared error 58.7045 %
- Total Number of Instances 9235
-

No POS Results

=== Stratified cross-validation ===

- === Summary ===
-
- Correctly Classified Instances 7539 81.6351 %
- Incorrectly Classified Instances 1696 18.3649 %
- Kappa statistic 0.8125
- Mean absolute error 0.0009
- Root mean squared error 0.0259
- Relative absolute error 19.3179 %
- Root relative squared error 55.1739 %
- Total Number of Instances 9235

No Context Results

=== Stratified cross-validation ===

- === Summary ===
-
- Correctly Classified Instances 7595 82.2415 %
- Incorrectly Classified Instances 1640 17.7585 %
- Kappa statistic 0.8187
- Mean absolute error 0.0008
- Root mean squared error 0.0255
- Relative absolute error 18.8978 %
- Root relative squared error 54.1967 %
- Total Number of Instances 9235

References

Berman, R. A. (1990). Acquiring an (S)VO language: Subjectless sentences in children's Hebrew. Linguistics, 28.

Berman, R. A. (2003). Children's lexical innovations. Language Acquisition and Language Disorders, 28.

Daya, E., Roth, D., & Wintner, S. (2008). Identifying Semitic Roots: Machine Learning with Linguistic Constraints Computational Linguistics 34(3).

Frost, R., Deutsch, A., Gilboa, O., Tannenbaum, M., & Marslen-Wilson, W. (2000). Morphological priming: Dissociation of phonological, semantic, and morphological factors. Memory & Cognition, 28.

MacWhinney, B. (2000). The CHILDES Project: Tools for analyzing talk. Third Edition. Mahwah, NJ: Lawrence Erlbaum Associates.