* In this paper, we try to maximize accuracy of these vector operations by developing new model architectures that preserve the linear regularities among words. We design a new comprehensive test set for measuring both syntactic and semantic regularities.
* The quality of these representations is measured in a word similarity task, and the results are compared to the previously best performing techniques based on different types of neural networks.
* Two new model architectures were introduced for learning distributed representations of words that try to minimize computational complexity. The main observation was that most of the complexity is caused by the non-linear hidden layer in the previous models. While this is what makes neural networks so attractive, we decided to explore simpler models that might not be able to represent the data as precisely as neural networks but can possibly be trained on much more data efficiently.
* The new architectures directly follow those proposed in our earlier work, where it was found that neural network language model can be successfully trained in two steps: first, continuous word vectors are learned using simple model, and then the N-gram NNLM is trained on top of these distributed representations of words.
* Continuous Bag-of-Words Model:
  + Similar to the feedforward NNLM, where the non-linear hidden layer is removed and the projection layer is shared for all words (not just the projection matrix); thus, all words get projected into the same position (their vectors are averaged).
  + Training complexity is then
    - * + Q = N x D + D x log2(V)
  + unlike standard bag-of-words model, it uses continuous distributed representation of the context.
* Continuous Skip-gram Model:
  + Similar to CBOW, but instead of predicting the current word based on the context, it tries to maximize classification of a word based on another word in the same sentence. More precisely, we use each current word as an input to a log-linear classifier with continuous projection layer, and predict words within a certain range before and after the current word.
  + The training complexity of this architecture is proportional to
    - * + Q = C x (D + D x log2(V ))