Deep Neural Network for Self-taught Learning

Evan Xiangwen Liu, Tolgahan Cakaloglu, Micah Hughes

Advised by: Xiaowei Xu

University of Arkansas at Little Rock

Address: 2801 S University Ave, Little Rock, AR 72204.

Abstraction

We present a novel machine learning framework to transfer hidden features from unlabeled text data based on distributed word representation.

Different from semi-supervised learning, which uses additional unlabeled instances following the same class labels or generative distributions as the labeled data.
And different from transfer learning, which uses labeled datasets to enhance classification.

We use a large amount of randomly downloaded data without a label to improve performance on our given datasets. Every document is represented as a sequence of vectors to feed in (CNN) Convolutional Neural network for classification.

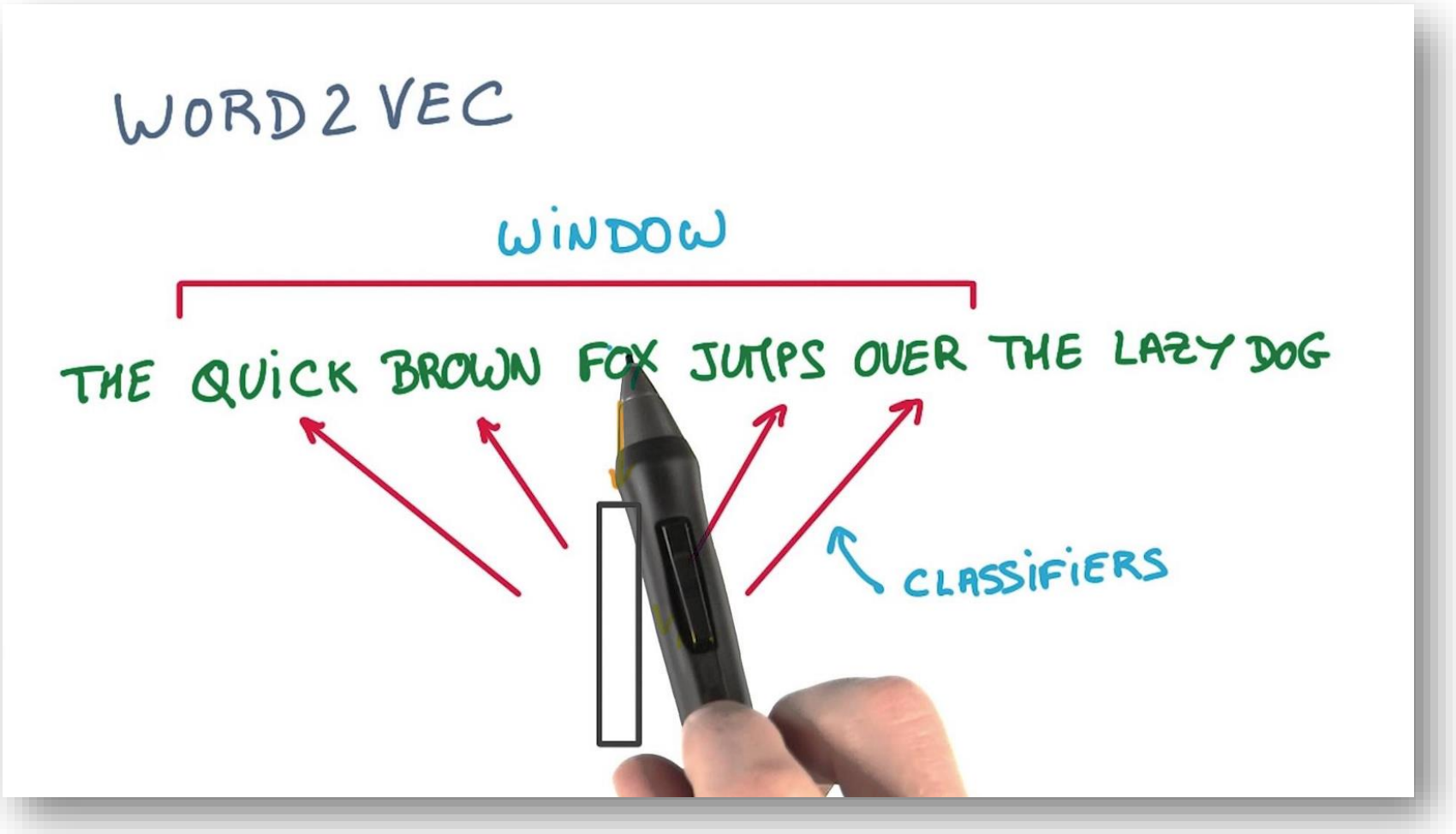

Figure 1 Skip-gram model from becominghuman.ai

Introduction

In deep neural networks, one of the most reliable ways to get better performance is to give the algorithm more data. This has led to the aphorism that in machine learning, “sometimes it’s not who has the best algorithm that wins; it’s who has the most data.”

Although there are lots of text data online, trying to get more labeled data is expensive, thus the algorithm has to get the ability to extract significant features from unlabeled data. In particular, the promise of self-taught learning and unsupervised feature learning is that if we can get our algorithms to learn from ”unlabeled” data, then we can easily obtain and learn from massive amounts of it.

Even though a single unlabeled example is less informative than a single labeled example, if we can get tons of the former—for example, by downloading random unlabeled text documents off the internet and if our algorithms can exploit this unlabeled data effectively, then we might be able to achieve better performance than the massive hand-engineering and massive hand-labeling approaches.

Method: Learning Representation

As a core task of NLP (Natural language Processing), document representation is an interesting and challenging task which is concerned with representing textual documents in a vector space, and it has various applications in text processing, retrieval and mining. As in (Figure 2), words are represented by vectors in vector space of certain dimensionality.

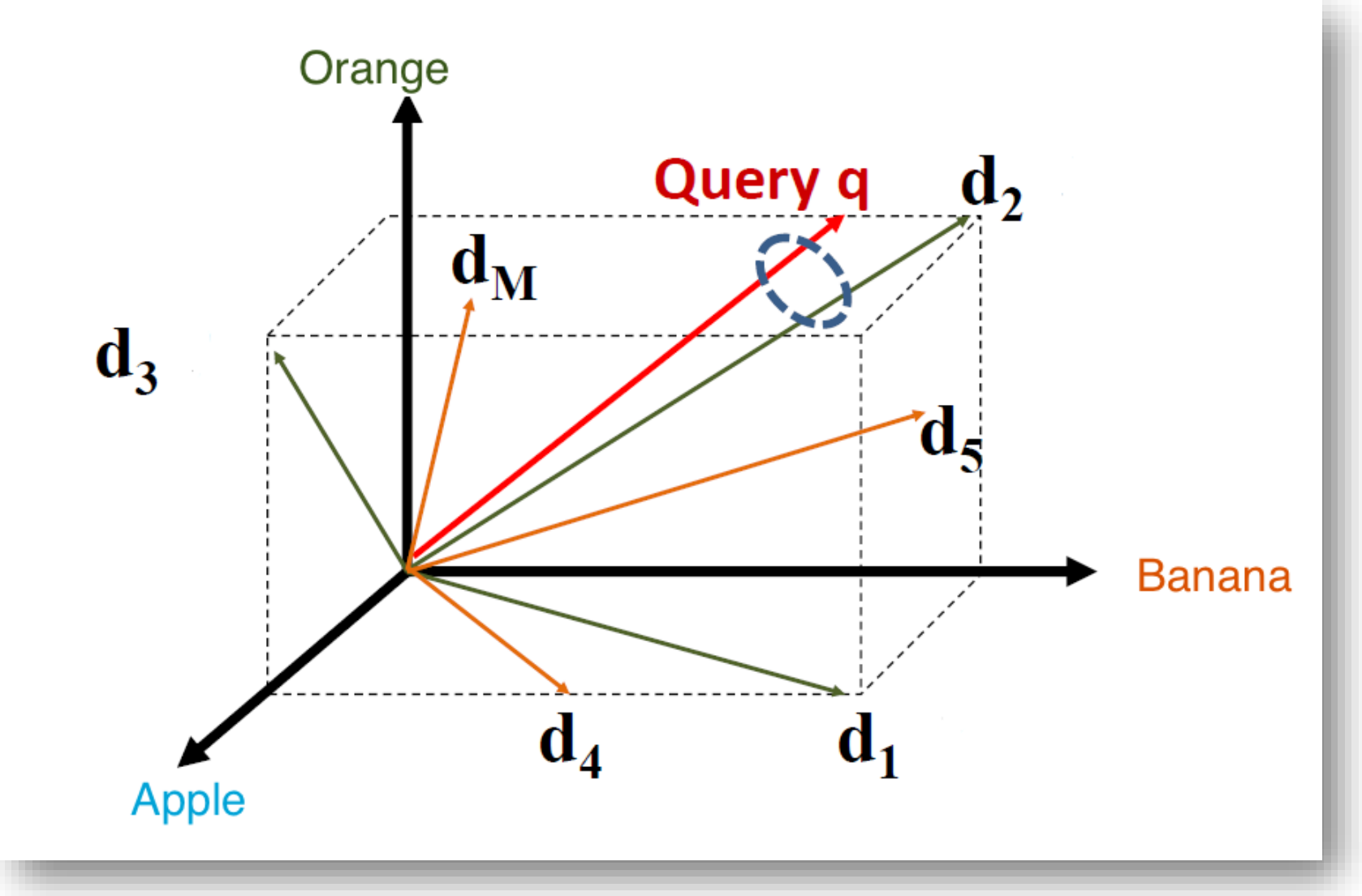

Figure 2 Words in vector space

Many current NLP systems and techniques treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. However, the simple techniques are at their limits in many tasks. The recently introduced continuous Skip-gram model (Figure 1) is an efficient method for learning high-quality distributed vector representations that capture a large number of precise syntactic and semantic word relationships.

Distributed representations of words in a vector space help learning algorithms to achieve better performance in natural language processing tasks by grouping similar words. The training layers of Skip gram model is shown in (Figure 3).

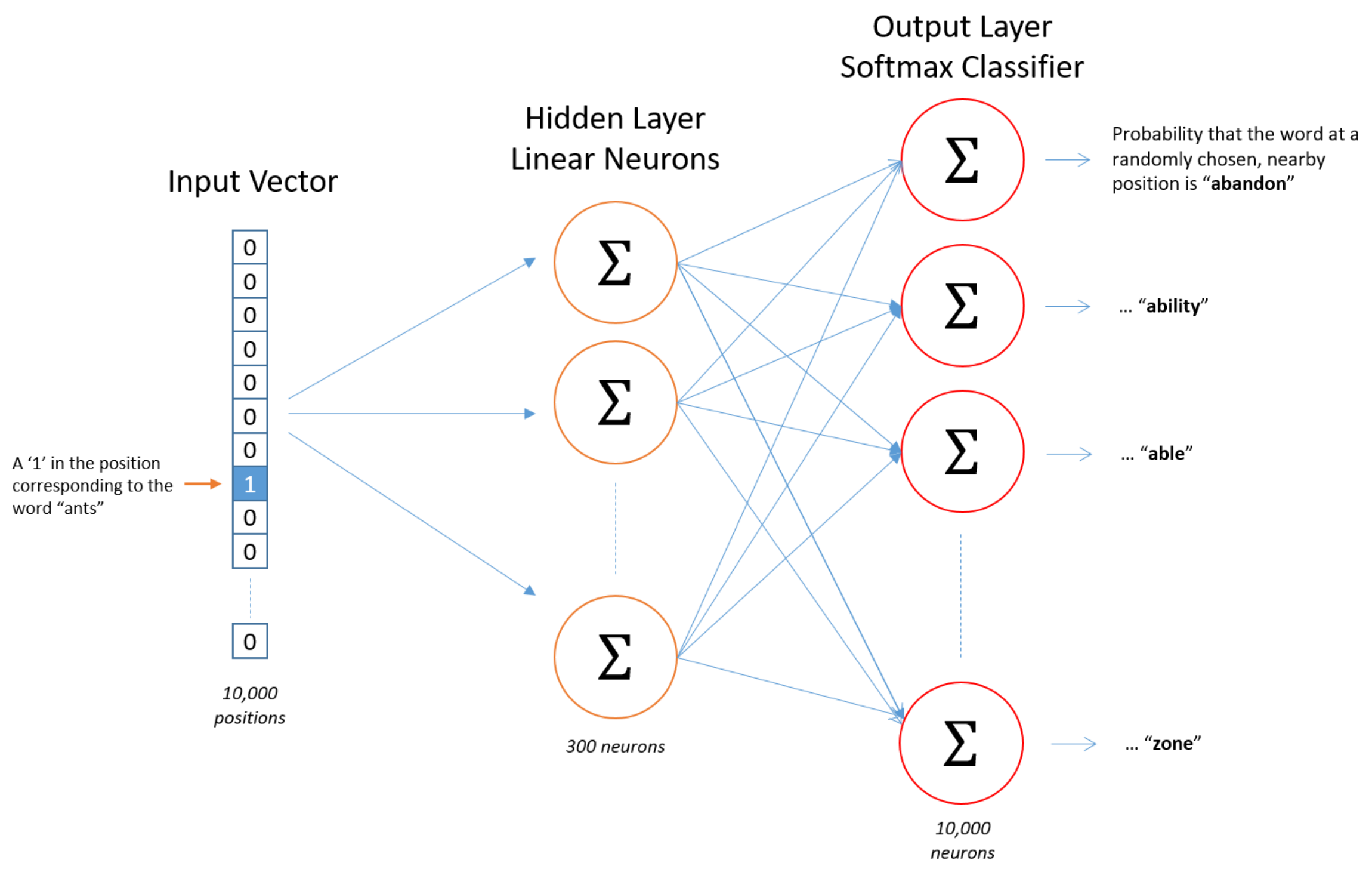

Figure 3 Training Layers of Skip-gram model from Chris McCormick Word2Vec Tutorial

Method: Convolutional Neural Network

In classification of labeled datasets, we use CNN(Convolutional Neural Network) to classify out datasets. All the layers is shown in (Figure 4).

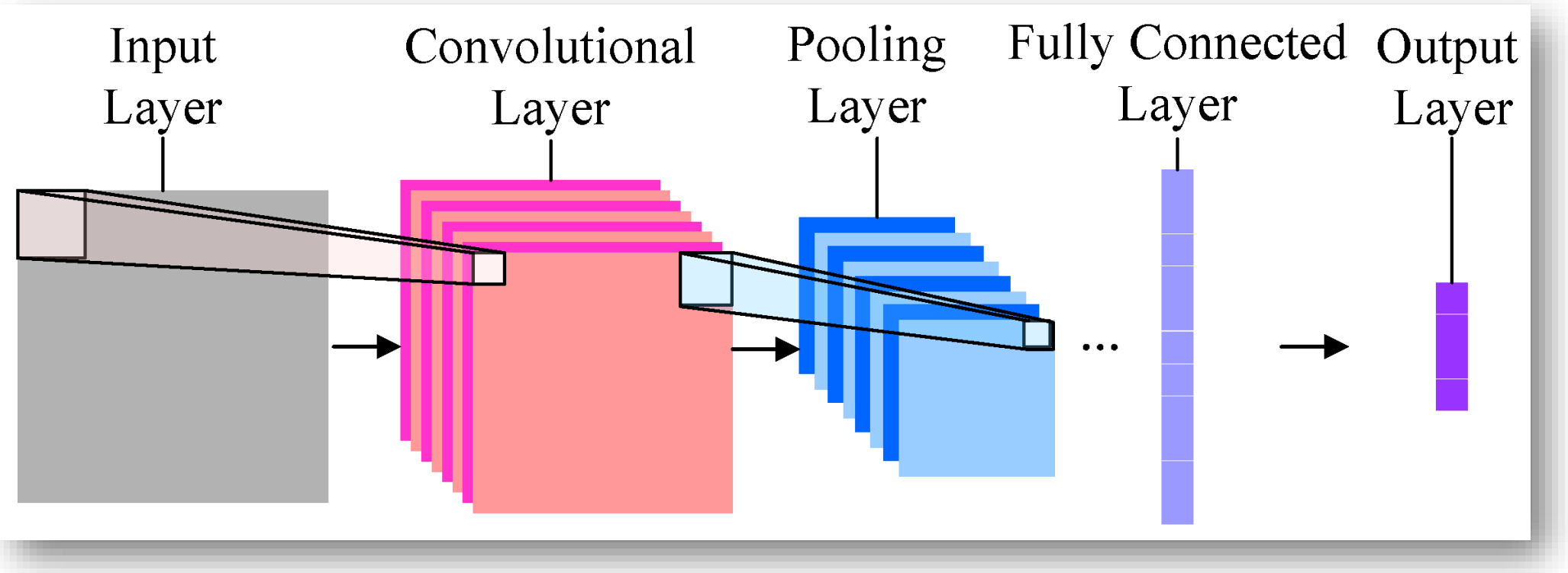

Figure 4 Layers of CNN(Convolutional Neural Network)

In our model architecture, the model is shown in Figure 5. Let $x_i \in R^k$ be the k-dimensionalword vector corresponding to the i-th word in the sentence. (Yoon Kim et al., 2014). A sentence of length n (padded where necessary) is represented as

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n,$$

where $\oplus$ is the concatenation operator. In general,let $x_{i:i+j}$ refer to the concatenation of words $x_i, x_{i+1}, \ldots, x_{i+j}$. A convolution operation involves a filter $w \in R^{hk}$, which is applied to a window of h words to produce a new feature. For example, a feature c_i is generated from a window of words $x_{i:i+h-1}$ by

$$c_i = f(w \cdot x_{i:i+h-1} + b).$$

Here $b \in R$ is a bias term and f is a non-linear function such as the hyperbolic tangent. This filter is applied to each possible window of words in the sentence $\{x_{1:h}, x_{2:h+1}, \ldots, x_{n-h+1:n}\}$ to produce
a feature map
$c = [c_1, c_2, \ldots, c_{n-h+1}]$, with $c \in R^{n-h+1}$.

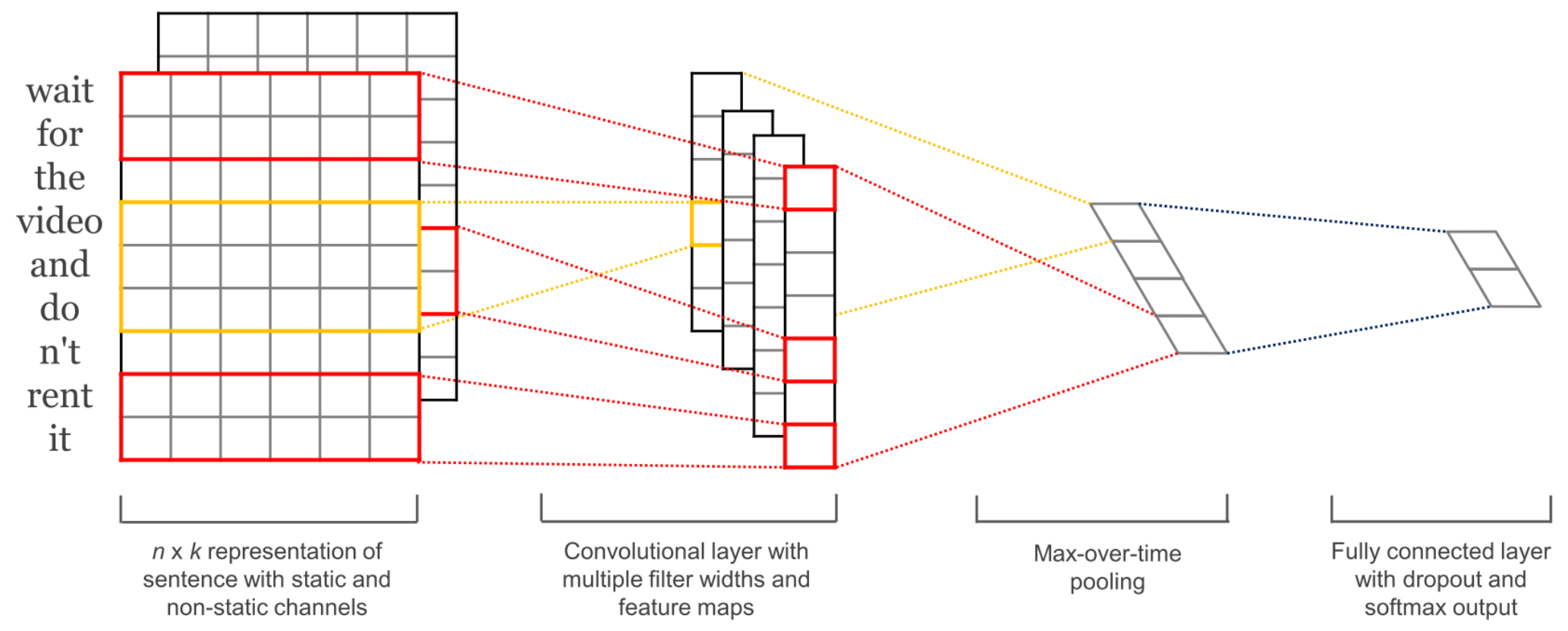
wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 5 CNN model architecture from Yoon Kim et al., 2014

Results

We tested our model on the following four datasets, and compared the results from different embedding: results are shown in Figure 6.

- **BBC News datasets:**
  2225 documents from 2004-2005.
- **Cornell University Movie review datasets:**
  1000 positive and 1000 negative movie reviews.
- **Stanford University Large Movie Review datasets:**
  50,000 binary movie reviews.
- **Amazon Food Review datasets:**
  ~500,000 fine food reviews in period of 10 years.

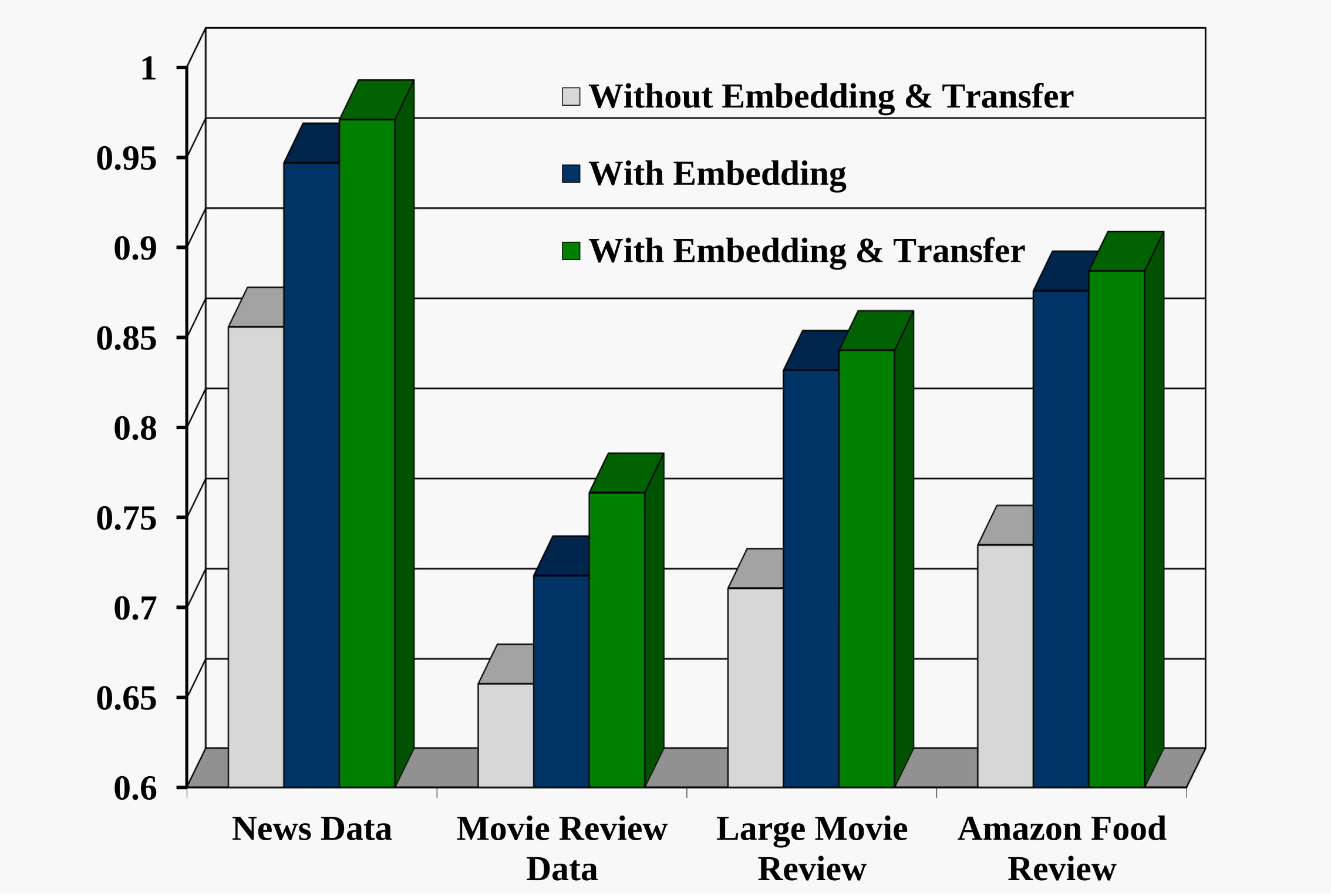

Figure 6 Tests results for different datasets

Conclusion

Based on the tests from different datasets, (as shown in Table 1)with embedding or without embedding, we can get the following conclusion:

- Our model outperforms the original Convolutional neural network for classification without embedding.
- Self-taught learning for using unlabeled data extracted important features.
- Sematic-Syntactic word relationships learned from skip gram model enhanced our supervised learning.

Tests results for different datasets				
	News Data	Movie Review Data	Large Movie Review	Amazon Food Review
Without Embedding & Transfer	0.856	0.658	0.711	0.735
With Embedding	0.947	0.718	0.832	0.876
With Embedding & Transfer	0.971	0.764	0.843	0.887

Table 1 tests results from different datasets