Image Understanding: Learning from Text

Miao Liu
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
maiol1@andrew.cmu.edu

Wentai Zhang
Department of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
wentaiz @andrew.cmu.edu

Zeyu Peng
Tepper school of business
Carnegie Mellon University
Pittsburgh, PA 15213
zeyup@andrew.cmu.edu

Abstract

This paper presents approaches of extracting nouns or pronouns from image descriptions. Image semantic segmentation will be accomplished with coarse tuned Full Convolutional Network (FCN. We also propose a learning from text algorithm to improve the segmentation performance when object has similar color to the background.

1 Introduction

Due to the improvement of computational capability, research in the field of computer vision starts to solve complicated computer vision problems, like image recognition or even image understanding. Image understanding has major application in many fields, for instance unmanned vehicle. One central idea of image understanding is to collect information from a certain image as much as possible, like what kind of objects are in this image, what are their numbers, and what are their geometry relationship. The foundation of image understanding is semantic segmentation. Previous works [1, 2, and 3] have used convolutional neural network for semantic segmentation. Aligning descriptions with images is also important for image understanding. Since texts can provide additional information, aligning texts with image can extract meaningful objects from an image and improve the performance of image segmentation. In this paper, we propose a method that uses both image descriptions and image features to generate Markov Random Process. This method can give a more detailed image segmentation result.

2 Related work

There has been substantial work in automatic caption or description generation of images [4, 5, and 6] and video [7]. The work enlightens us is [8], where short sentences are parsed into nouns and prepositions, which are used to generate potentials in a holistic scene model. Only a few datasets contain images and text. For POS tagging, the most outstanding work has been done by the Stanford NLP group and they implement maximum entropy [9] and Feature-rich [10] in POS. Then a hidden Markov Model has been used in [11], where they consider the joint probability of a word chain when judging the word tag. Also, another successful model Tree Tagger was created using decision tree [12] and it also works well on German [13]. Very little work has been devoted to exploiting text to improve semantic visual understanding beyond simple image classification The UIUC dataset [14] augments a subset of PASCAL VOC 08 with 3 independent sentences (each by a different annotator) on average per image.

3 Methods

3.1 Semantic Segmentation

Semantic segmentation is accomplished based on Fully Convolutional Network. Its structure is shown in figure1.

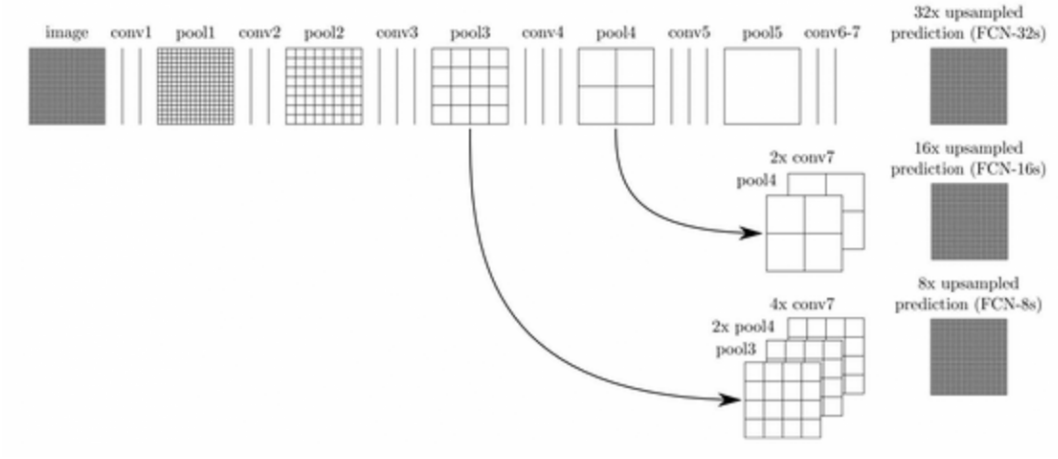

Figure 1. Fully convolutional network structure

We constructed this network with Matconvnet. Matconvnet is a MATLAB toolbox designed for implementing convolutional neural network for computer vision applications.

3.2.1 Natural Language Process

In regards to text NLP part of our project, our goal falls into the category of part of speech (POS) tagging where we would like to implement a systematic algorithm which is able to identify the parts of speech of each word in a simple sentence. For example, given the description 'there are five apples on the table', there are two nouns – apples and table, one number – five, and one preposition – on. By understanding there are two main objects to extract where the number of one of them is five and the apples are actually above the table, our image segmentation algorithm will be able to extract the useful objects from the image.

Our current implemented model is a hidden Markov model (HMM) based on maximum likelihood. Suppose that we have n words in sentence, $w_1, w_2, .., w_n$, we would like to assign them tags $t_1, t_2, \ldots t_n$ such that

$$(\widehat{t_1}, \widehat{t_2}, \ldots, \widehat{t_n}) = \text{argmax}_{t_1,t_2,\ldots,t_n} P(t_1, t_2, \ldots, t_n | w_1, w_2, \ldots, w_n)$$

which is the probability of the tag sequence conditioning on observing the words $w_1, w_2, .., w_n$. Bayes' theorem tells us that

$$P(t_1, t_2, \ldots, t_n | w_1, w_2, \ldots, w_n) = \frac{P(\bar{w}|\bar{t})P(\bar{t})}{P(\bar{w})}$$

where $\bar{w}$ and $\bar{t}$ are vectors of words and tags. Here we would like to make some simplifying and rather reasonable assumptions: The probability of a particular word depends only on its corresponding tags but not on other tags, that is $P(\bar{w}|\bar{t}) = \prod P(w_i|t_i)$. The probability of a tag only depends on the tag of previous k tags, that is $P(\bar{t}) = \prod P(t_i|t_{i-1}, t_{i-2}, .., t_{i-k})$.

In our implementation, we chose k = 3, that is each tag only depends on its previous two tags in the sentence. To train this model, we used a corpus of labelled words with over 200K sentences. To estimate the probabilities in the model, we use MLE. Denote the counting of occurrence as a function C, we have

$$P(t_i|t_{i-1}) = \frac{C(t_i, t_{i-1})}{C(t_{i-1})}$$

which is the number of occurrence of two consecutive tags t_{i-1}, t_i divided by the number of occurrence of tag t_{i-1} alone in the training texts. Similarly, we estimate $P(w_i|t_i)$ by $P(w_i|t_i) = \frac{C(w_i,t_i)}{C(t_i)}$, that is out of all the occurrence of tag i, what is the proportion this tag is associated with word i.

3.2.2 Neural Network

The second model we proposed is based on a shallow neural network which also looks at the previous word in the sentence. The input of the network consists of 45 variables. The first 44 variables represent the probability of the word being each of the 44 possible POS tag as in the training set, while the last input is the POS tag of the previous word in the sentence which is represented by the number of 0-45 where number 0 means this word is the beginning of a sentence. For the hidden layer, the activation function we used is sigmoid function given by $\sigma(x) = \frac{1}{1+e^{-x}}$. For the output layer, we have 44 outputs, which corresponds to the probability of the word being each POS tag in the that sentence. Suppose applying weight matrix to the hidden layer yields a 44×1 vector I, then the softmax activation function has the form $O_i = \frac{e^{I_i}}{S}$ where $S = \sum e^{I_i}$. The POS tag with the largest probability is then chosen and later used as input for the next word in the sentence. The loss function for this model would be $L = \frac{1}{N}\sum L_i + \frac{\lambda}{2}\sum_{l=0}^{nh}\sum_d\sum_f\left(W_{ij}^{(l)}\right)^2$ where nh is the number of hidden layer which is one in our model, λ is regularization parameter and L_i is the loss function for each individual training data and is given by $L_i = -\log\frac{e^{y_i}}{s}$ where y_i is the true observed POS tag of the training data. Like the HMM model above, for words that do not appear in training data, we could not have any prior probabilities for the word being each POS tag. In this case, we give equal probability of each POS tag and would heavily rely on the previous tag in the sentence. A graphical explanation of the model is as follow:

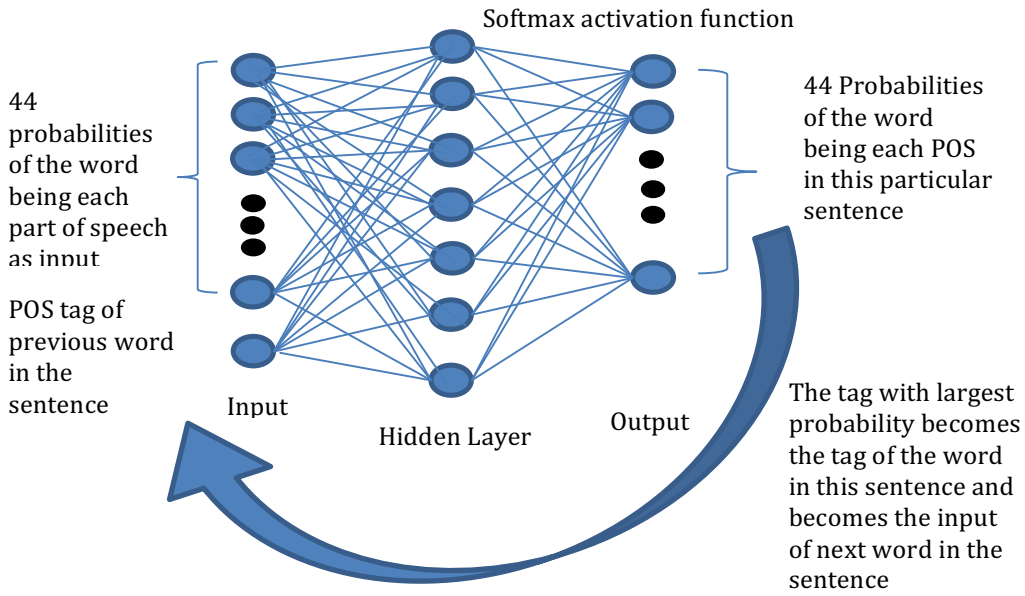

Figure 2. NLP Neural Network structure

3.3 Learning form Text Algorithm

The output of segmentation is a mxnx21 matrix. (m, n correspond to image height and width) The output of NLP is the nouns of image description and corresponding category of PASCAL

Dataset. We define a Markov Random Field (MRF) potential function which contains information of image segmentation, as well as each noun/pronoun of interest to which visual concept corresponds. The work flow of our algorithm is shown in the following image:

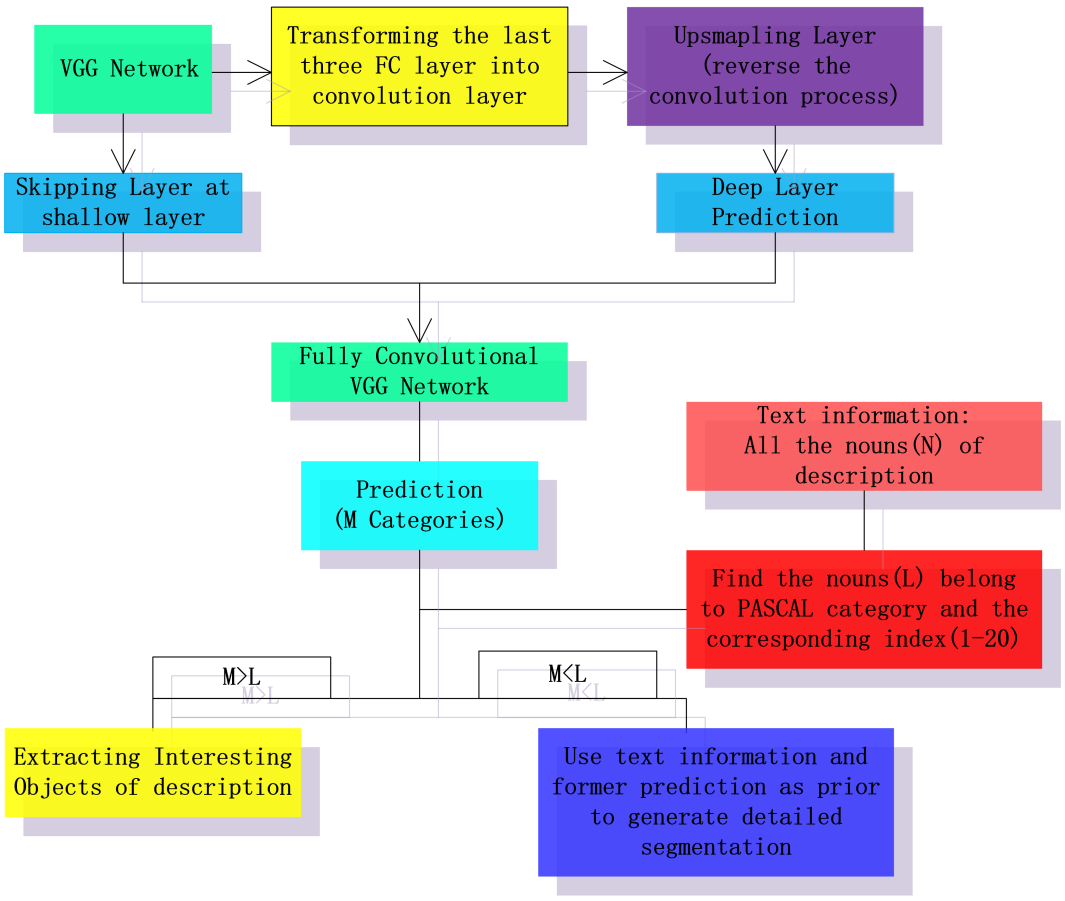

Figure 3. Learning form Text Algorithm structure

4 Dataset

For image segmentation, we will use PASCAL VOC 2012 Dataset, which is the most famous dataset for computer vision task like object detection, localization which refers to detecting the object instance in the image, and providing the bounding box coordinates around the object instance or segmenting the object. PASCAL has its own validation dataset for image segmentation. As for learning from text algorithm, we use the 'cat on chair' image for evaluation.

For POS tagging, we used CoNLL-2000 Chunking Dataset. We used the *train.txt* and *test.txt* for training and testing. In the dataset, the format is that every row includes a word, a tag and an indicating tag. The word in each row can form a complete sentence. Since we only want to know the tag, we used the first two column of the dataset as observations and corresponding labels. There are 44 different types of tags in total. And in training set, there 211727 words and 17260 unique words. In test set, there are 47377 words in total.

We also did some preprocessing to training set. Because we need to find the word pairs for the further tagging process, we extract all existing word pairs existing in the training set. There totally 1131 two-word pairs appeared in the training set while the total possible pairs would be 1936. About 800 pair features are missing but I think most of the missing features may be because there are no such grammar structures in the language formation principles.

5 Result

5.1 Semantic Segmentation

The accuracy of semantic imagination will be evaluated by the mean accuracy of validation dataset from PASCAL VOC 2012. The mean accuracy in 75.8%. The visualized confusion matrix of validation set. The visualization of segmentation confusion matrix is shown in figure 4

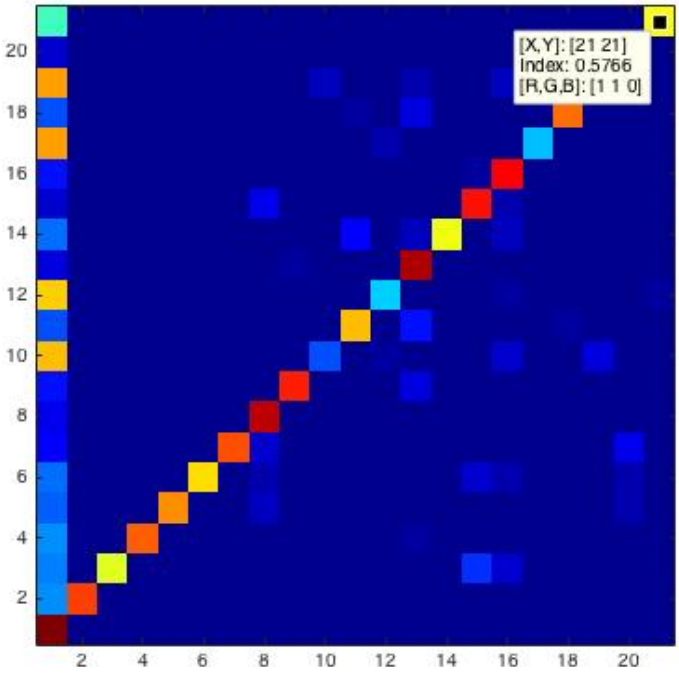

Figure 4. Semantic segmentation with pretrained CNN

5.2.1 Hidden Markov Model

For parsing the description, we implement the method described above. Because the speed of MATLAB on reading and clustering is really slow. We use python to read and parse the word pairs. And in MATLAB, we test the performance on training data and test data or even a command window sentence input. The accuracy is obtained by comparing with true label.

For Hidden Markov model, the training accuracy is 94.92% and test accuracy is 89.59%. As an example for parsing an arbitrary command window input, we tried “There is a man in the door. He is old.” And the tag outcome is “'EX' 'VBZ' 'DT' 'NN' 'IN' 'DT' 'NN' 'PRP' 'VBZ' 'JJ'”. This is correct. And the result is correct for most common sentences.

Name	Accuracy(%)
Km01[15]	93.45
Osb00[16]	91.65
Vd00[17]	88.82
Joh00[18]	86.24
Ours	89.59

Table1.Comparison on accuracy

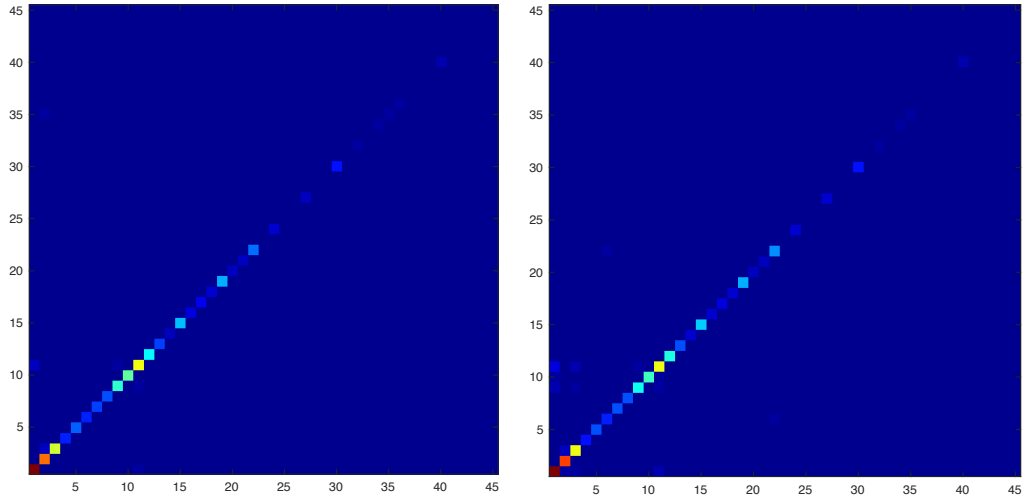

Figure 5. (a) the confusion matrix for the training set. (b) the confusion matrix for the test set

Test	There	is	a	cat	on	the	chair
Label	EX	VBZ	DT	NN	IN	DT	NN

Table 2. An example test for tagging an image description. Two nouns are detected.

5.2.2 Neural Network

The training accuracy and test accuracy of NN is not as good as HMM model with accuracy below 50% for a variety of parameters including learning rate, number of hidden layer units, regularization parameter. Something we should notice is that this network is converging to a minimum point for the loss function as shown in the following plot. However, we highly suspect that the network was stuck in a local minimum which results in a not so promising result.

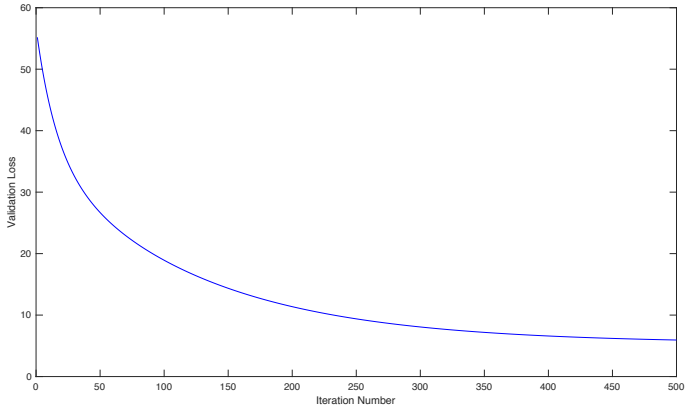

Figure 6 Loss function plot($\lambda = 0.4, \alpha = 0.01, h_1 = 200$)

6 Discussion and Analysis

6.1 POS tagging with HMM

As we can see, the HMM model achieved a very promising accuracy compared with previous works that have been done on the same dataset. Judging from the two confusion matrix in figure 3, we can find that there are rare situations that the model failed to tag the word (the 45^{th} column). Because we are now using 3-word pair in the model. The only situation when a

tag can't be made is when there is no such word in the vocabulary and there is no such 2-word and 3-word pair in the preprocessed data. This hardly happens. In fact, the word 'cat' and 'chair' are not in the vocabulary. But we can still achieve correct tags by the word pairs.

By scanning the tagging result of our test data, we can see that for most of the errors, the cause is the misclassification between different types of nouns (There four types of nouns), especially between a noun and its own plural form. I think this can be improved if we can add a new possibility into consideration, which is judging if the word is plural by its suffix given that this word has been tagged a noun. We can collect common plural suffix such as 's', 'es', 'ies' and etc.

Another limitation is the first word in each sentence. Because we cannot use 3-word pairs. We can only use 2-word pair with the former element ' ' (blank) to judge the tag when the word itself is not in the vocabulary. This is unfair because every word in this situation will be given the same tag which is 'DT' in our algorithm. A simple and promising way to improve this will be add a backward pass for the first word in sentence. Once the second word has been tagged in the sentence. We can check the word pair with the latter one fixed and decide the tag for the first one again. But this should only be implemented when the first word is not in the vocabulary.

6.2 POS tagging with NN

After some investigation, we have conjectured some possible reason for the underperformance of the NN and hopefully we could improve some these points in future implementation. First, original data is very unbalanced with some of the POS tag occurring less than 10 times out of 210K words. To counter this problem, we can try to upsample or downsample the training data or simply give weights to the loss function to balance the data. Another possible cause of the underperformance is that out of 45 input variables, the first 44 are probabilities ranging from 0 to 1 while the last one is integer from 0 to 44. The last input's integer has no numerical meaning at all but still has a rather wide range. Tag 44 should not be numerically much greater than tag 1, but the NN may not understand this since it simply does computation with weight matrix. Hence, this might drastically reduce the performance of the NN. To solve this problem, we suggest that, in future implementation, we should use a 44 binary input to represent the POS tag of the previous tag instead of single number.

6.3 Learning from Text Algorithm Result:

As we can see from image segmentation confusion matrix, one major problem is that many pixels are wrongly classified as background (category 1). Take the 'cat on chair' image (Figure 7(a)) as an example, it's the mean accuracy is only 9.1%. Many interesting pixels (chair or bicycle), are classified as background, which greatly compromised the segmentation performance. We use 'There is a cat on the chair. 'as image description. The interesting nouns (cat, chair) correspond to PASCAL category 8 and 9. Then we use this priori and original FCN image segmentation result to conduct our learning from text algorithm. As shown in figure 7(b), our Learning from Text Algorithm shows a great improvement on segmentation performance. For this single image, the mean pixel accuracy improved from 9.1% to 14.7%.

(a) (b) (c)

Figure 7 Baseline segmentation result, Learning from Text result and ground truth

References

[1] D.C.Ciresan,A.Giusti,L.M.Gambardella,andJ.Schmid- huber. Deep neural networks segment neuronal membranes in electron microscopy images. In NIPS, pages 2852–2860, 2012. 1, 2, 4, 7

[2] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013. 1, 2, 4, 7,8

[3] B. Hariharan, P. Arbela´ez, R. Girshick, and J. Malik. Simul- taneous detection and segmentation. In European Confer- ence on Computer Vision (ECCV), 2014. 1, 2, 4, 5, 7, 8

[4] G.Kulkarni,V.Premraj,S.Dhar,S.Li,Y.Choi,A.Berg,and T. Berg. Baby talk: Understanding and generating simple image descriptions. In CVPR, 2011. 1, 2

[5] P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi. Collective generation of natural image descriptions. In Asso- ciation for Computational Linguistics (ACL), 2012. 2

[6] P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi. Generalizing image captions for image-text parallel corpus. In Association for Computational Linguistics (ACL), 2013.

[7] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickin- son, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, L. Schmidt, J. Shangguan, J. Siskind, J. Waggoner, S. Wang, J. Wei, Y. Yin, and Z. Zhang. Video-in-sentences out. In UAI, 2012. 1, 2

[8] S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In CVPR, 2013. 1, 2

[9] Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70.

[10] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

[11] Speech and Language Processing. Daniel Jurafsky & James H. Martin. 2014. Chapter 9.

[12] Helmut Schmid (1994): Probabilistic Part-of-Speech Tagging Using Decision Trees. Proceedings of International Conference on New Methods in Language Processing, Manchester, UK.

[13] Helmut Schmid (1995): Improvements in Part-of-Speech Tagging with an Application to German. Proceedings of the ACL SIGDAT-Workshop. Dublin, Ireland.

[14]A.Farhadi,M.Hejrati,M.Sadeghi,P.Young,C.Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences for images. In ECCV, 2010. 1, 2

[15] Taku Kudoh and Yuji Matsumoto, Use of Support Vector Learning for Chunk Identification. In: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal, 2000.

[16] Miles Osborne, Shallow Parsing as Part-of-Speech Tagging. In: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal, 2000.

[17] Jorn Veenstra and Antal van den Bosch, Single-Classifier Memory-Based Phrase Chunking. In: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal, 2000.

[18] Christer Johansson, A Context Sensitive Maximum Likelihood Approach to Chunking. In: Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal, 2000.