Midge: Generating Descriptions of Images[*]

Margaret Mitchell
University of Aberdeen
m.mitchell@abdn.ac.uk

Xufeng Han
Stony Brook University
xufhan@cs.stonybrook.edu

Jeff Hayes
SignWorks of Oregon
jeff@signworksoforegon.com

Abstract

We demonstrate a novel, robust vision-to-language generation system called Midge. Midge is a prototype system that connects computer vision to syntactic structures with semantic constraints, allowing for the automatic generation of detailed image descriptions. We explain how to connect vision detections to trees in Penn Treebank syntax, which provides the scaffolding necessary to further refine data-driven statistical generation approaches for a variety of end goals.

1 Introduction

There has been a growing interest in tackling the problem of how to describe an image using computer vision detections. This problem is difficult in part because computer vision detections are often wrong: State-of-the-art vision technology predicts things that are not there, and misses things that are obvious to a human observer. This problem is also difficult because it is not clear what kind of language should be generated – the language that makes up a "description" can take many forms.

At the bare minimum, an automatic vision-to-language system, given an image with a single detection of, for example, a dog, should be able to generate *a dog*, and a longer phrase if requested. To be useful in real-world applications, it should be able to create basic descriptions that are as true as possible to the image, as well as descriptions that guess probable information based on language analysis alone. To our knowledge, no current system provides this functionality. Midge is built based on these goals.

Our approach converts object detections to descriptive sentences using a tree-generating derivation process that fleshes out lexicalized syntactic structure around object nouns. Likely subtrees are learned from a cleaned version of the Flickr dataset (Ordonez et al., 2011) parsed using the Berkeley parser. The final structures generated by the system are present-tense declarative sentences in Penn Treebank syntax.

With this in place, the system can generate *a dog*, *a black dog sleeping*, *a furry black dog sleeping by a cat*, etc., while also suggesting further detectors for the vision system to run. Approaching the problem in this way, Midge provides a starting point for generation to meet different goals: from automatically creating stories or summaries based on visual data, to suggesting phrases that a speech-impaired AAC user can select to assist in conversation. There is still much work to be done, but we believe that the basic architecture used by this system is a solid starting point for generating a wide variety of descriptive content, and makes clear some of the issues a vision-to-language system must handle in order to generate natural-sounding descriptions.

2 Background

Previous work on generating image descriptions can be characterized as prioritizing among several goals:

- Creating language that is poetic or metaphorical (Li et al., 2011)
- Creating automatic captions with syntactic variation based on semantic visual information (Farhadi et al., 2010)
- Creating language describing the scene in a basic template-driven way, utilizing attribute detections (Kulkarni et al., 2011) or likely verbs from a language model (Yang et al., 2011)

To meet one goal, other goals are often compromised. Yang et al. (2011) fill in likely verbs to form complete sentences, but limit the generated structures to a simple template, without capturing natural variation in sentence length or surface structure.

[*] Thanks to the CLSP 2011 summer workshop at Johns Hopkins for making this system possible. Midge is available to try online at http://recognition.cs.stonybrook.edu:8080/˜mitchema/midge/ and http://mcvl.cewit.stonybrook.edu//˜mitchema/midge/ and screenshots at http://www.abdn.ac.uk/˜r07mm9/midge/

INLG 2012 Proceedings of the 7th International Natural Language Generation Conference, pages 131–133,
Utica, May 2012. ©2012 Association for Computational Linguistics

Li et al. (2011) aim at more metaphorical and varied language, but the generated structures are often syntactically and semantically ill-formed. Farhadi et al. (2010) generate natural, varied, descriptive language, but this is created by copying captions directly from similar images, resulting in captions that are often not true to the actual query image content.

Midge builds on ideas from these systems, additionally mapping the structures underlying vision detections to syntactic structures and data-driven distributional information underlying natural language descriptions. With this in place, the door is opened for language and vision to communicate at a deep syntactic-semantic level. The language components of the system can filter and expand on given visual information, and can also call back to the visual system itself, specifying further detectors to run (or train) based on semantically related or expected information. We hope that this system not only advances work in generating visual descriptions, but work in training visual detectors as well.

3 Vision to Language Issues

The process of developing Midge brought to light several key issues that any vision-to-language system aiming to generate descriptive, varied, human-like language must handle:

Descriptiveness: Should the system include information about everything there is evidence for, limit that information, or add to it?

World knowledge: What sorts of things in an image are remarkable, and should be mentioned, and which may go without saying?

Object grouping: Which objects should be mentioned together? How do people divide objects among sentences when they describe an image? Which detections should not be mentioned?

Noun ordering: In what order should the objects be named?

Reference plurals and sets: How should sets of objects be described as a whole? Should the exact number be included (*four chairs*), a vague term (*a few chairs*) or a general plural form (*chairs*)?

Modifier ordering: How should the different modifiers common to descriptions be ordered to make the utterances sound fluent?

Determiner selection: When should objects be treated as given (*the sky*), new (*a boy*), mass (*grass*), or count (*a blade*)?

Verb selection: Given that action/pose detection in computer vision does not function reliably, should verbs be hallucinated from a language model alone? Should they be left out?

Preposition selection: How should spatial relations between objects be analyzed, and how does this translate to language describing the scene layout?

Surface realization: What final lexicalization decisions need to be made to realize the generated strings within the output language?

Final string selection: Given a set of possible outputs, how is the final output string decided?

Nonsense detections: How should the system handle computer vision detections that are often wrong?

Many of these issues are well-suited to statistical NLP techniques, and some (modifier ordering, final string selection) have already been addressed in the NLP community. Where appropriate, Midge incorporates this technology alongside novel solutions to issues that have not yet been heavily researched (determiner selection, nominal ordering). We hope to further refine Midge's solutions as technology in these areas advances.

Separating Midge's architecture into components that handle each of these issues separately means that the system is flexible to change the kind of language it generates depending on the goals of the end user. The system offers general solutions to the issues listed above, and can have many of its goals changed if specified at run-time, resulting in different kinds of generated utterances. Midge can successfully create natural, varied descriptions that add descriptive content based on language modeling alone; it can also generate descriptions that are more limited, but as true as possible to the image.

4 Natural Language Generation in Midge

```
- id: 1, type: 1, label: bus, score: 0.73, bbox: [65.0, 65.0, 415.0,
191.0], attrs: {'blue': 0.01, 'furry':.02, . . . , 'shiny': 0.69}
- id: 2, type: 1, label: road, score: 0.95, bbox: [1.0, 95.0, 440.0,
235.0], attrs: {'blue': 0.01, . . .}
- preps {1,2}: 'by'
```

Figure 1: Computer Vision Out / Midge In (Excerpt)

The input to Midge is the output of vision detections, with detectors run for objects and attributes within each object's bounding box. In this demonstration, we incorporate the Kulkarni et al. (2011) vision detections. This provides objects/stuff and associated attributes, bounding boxes, and spatial relations between object pairs derived from the bounding boxes. Object detections are based on Felzenszwalb's multi-scale deformable parts models, and stuff detections are based on linear SVMs for low level region features.

Language generation in Midge is driven by a lexicalized derivation process that uses likely syntactic and distributional information for object nouns to create present-tense declarative sentences. Object detections form the basis of the computer vision detections, and these in turn are linked to nouns that form the basis of the generated output string.

The syntactic trees used to collect and generate likely subtrees for object nouns is outlined in Figure 2. Each anchor noun selects for a set of likely adjectives **a**, determiners **d**, prepositions **p** and present tense verbs **v**.

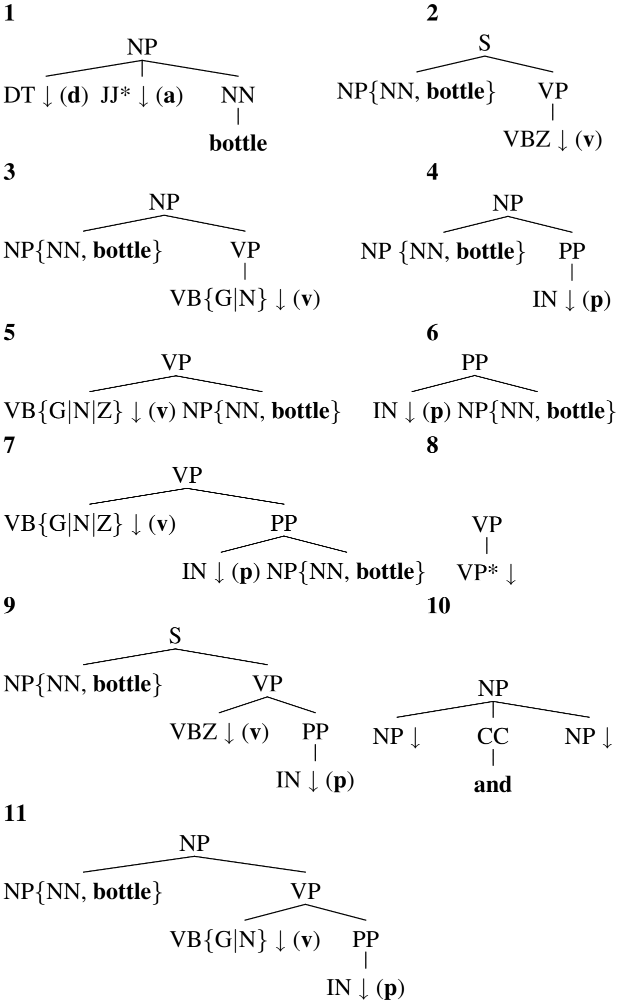

Figure 2: Trees for generation. Each {NN, **noun**} selects for its local subtrees. ↓ marks a substitution site, * marks ≥ 0 sister nodes of this type permitted. **Input:** set of ordered nouns, **Output:** trees preserving nominal ordering.

5 Architecture

Midge can be explained at a high level as a pipelined system incorporating the following steps:

Step 1: Run detectors for objects, stuff, action/pose and attributes; pass as <detection, score> pairs to Midge. Vision output/NLG input is displayed in Figure 1 and in the system demo.

Step 2: Group objects together that will be mentioned together.

Step 3: Order objects within each group – this automatically sets the subject and objects of the sentence. Midge currently order nouns based on WordNet hypernyms.

Step 4: Create all tree structures that can be generated from the object noun node. (See Figure 2). Noun anchors select for adjectives (JJ), determiners (DT), prepositions (IN) and if specified, verbs (VBG, VBN, or VBZ).

Step 5: Limit adjectives (JJ) to the set that are not *mutually exclusive* – different values for the same attribute class. REG comes into play at this step.

Step 6: Create all trees that combine following the given trees until all object nouns in a group are under one node (either NP or S).

Step 7: Order selected adjectives. We use the top-scoring ngram model from (Mitchell et al., 2011).

Step 8: Choose final tree from set of generated trees. Users can select a longest-string or cross entropy calculation.

References

A. Farhadi, M. Hejrati, P. Young Sadeghi, C. Rashtchian, J. Hockenmaier, and D. A. Forsyth. 2010. Every picture tells a story: generating sentences for images. *Proc. ECCV 2010.*

G. Kulkarni, V. Premraj, and S. Dhar, et al. 2011. Baby talk: Understanding and generating image descriptions. *Proc. CVPR 2011.*

S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. 2011. Composing simple image descriptions using web-scale n-grams. *Proc. CoNLL 2011.*

M. Mitchell, A. Dunlop, and B. Roark. 2011. Semi-supervised modeling for prenominal modifier ordering. *Proc. ACL 2011.*

V. Ordonez, G. Kulkarni, and T. L. Berg. 2011. Im2text: Describing images using 1 million captioned photographs. *Proc. NIPS 2011.*

Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. 2011. Corpus-guided sentence generation of natural images. *Proc. EMNLP 2011.*